2

RADC-TR-90-404, Vol VIII (of 18)
Final Technical Report
December 1990

ARTIFICIAL INTELLIGENCE APPLICATIONS TO SPEECH RECOGNITION

Northeast Artificial Intelligence Consortium (NAIC)

Harvey Rhody and John Biles

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

This effort was funded partially by the Laboratory Director's fund.

Rome Air Development Center
Air Force Systems Command
Griffiss Air Force Base, NY 13441-5700

This report has been reviewed by the RADC Public Affairs Division (PA) and is releasable to the National Technical Information Services (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-90-404, Volume VIII (of 18) has been reviewed and is approved for publication.

APPROVED: John G. Parker

JOHN G. PARKER
Project Engineer

APPROVED: Walter Senus

WALTER J. SENUS
Technical Director
Directorate of Intelligence & Reconnaissance

FOR THE COMMANDER: Igor G. Plonisch

IGOR G. PLONISCH
Directorate of Plans & Programs

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (IRAA) Griffiss AFB NY 13441-5700. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE December 1990	3. REPORT TYPE AND DATES COVERED Final Sep 84 - Dec 89

4. TITLE AND SUBTITLE
ARTIFICIAL INTELLIGENCE APPLICATIONS TO SPEECH RECOGNITION

5. FUNDING NUMBERS
C - F30602-85-C-0008
PE - 62702F
PR - 5581
TA - 27
WU - 13
(See reverse)

6. AUTHOR(S)
Harvey Rhody and John Biles

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Northeast Artificial Intelligence Consortium (NAIC)
Science & Technology Center, Rm 2-296
111 College Place, Syracuse University
Syracuse NY 13244-4100

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Rome Air Development Center (COES)
Griffiss AFB NY 13441-5700

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
RADC-TR-90-404, Vol VIII (of 18)

11. SUPPLEMENTARY NOTES (See reverse)
RADC Project Engineer: John G. Parker/IRAA/(315) 330-4024
This effort was funded partially by the Laboratory Director's fund.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The Northeast Artificial Intelligence Consortium (NAIC) was created by the Air Force Systems Command, Rome Air Development Center, and the Office of Scientific Research. Its purpose was to conduct pertinent research in artificial intelligence and to perform activities ancillary to this research. This report describes progress during the existence of the NAIC on the technical research tasks undertaken at the member universities. The topics covered in general are: versatile expert system for equipment maintenance, distributed AI for communications system control, automatic photointerpretation, time-oriented problem solving, speech understanding systems, knowledge base maintenance, hardware architectures for very large systems, knowledge-based reasoning and planning, and a knowledge acquisition, assistance, and explanation system.

The specific topic for this volume is the design and implementation of a knowledge-based system to read speech spectrograms.

14. SUBJECT TERMS
Artificial Intelligence, Expert Systems, Phoneme Classification, Speech Recognition, Signal Processing, Knowledge-Based Systems

15. NUMBER OF PAGES
56

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Block 5 (Cont'd)   **Funding Numbers**

PE - 62702F	PE - 61102F	PE - 61102F	PE - 33126F	PE - 61101F
PR - 5581	PR - 2304	PR - 2304	PR - 2155	PR - LDFP
TA - 27	TA - J5	TA - J5	TA - 02	TA - 27
WU - 23	WU - 01	WU - 15	WU - 10	WU - 01

Block 11 (Cont'd)

This effort was performed as a subcontract by the Rochester Institute of Technology to Syracuse University, Office of Sponsored Programs.

Table of Contents

Table of Figures and Tables

8.1 Executive Summary

Rochester Institute of Technology's (RIT) contribution to the Northeast Artificial Intelligence Consortium (NAIC) project has been the research and development of a system of techniques and processes suitable for use in a continuous speech, large vocabulary, speech understanding system. We have incorporated these techniques into an functioning workstation which is capable of testing, evaluating, and delivering this speech understanding technology. Other contributions made possible by this NAIC project have been: the engendering of AI capabilities within the Rome Air Development Center (RADC) & RIT, the support and educational growth of many students and researchers at RIT who worked on the project during the past five years, and some private industry involvement.

The research and development of RIT's speech understanding system carefully incorporated testing and evaluation methods at each level of planning, design, implementation, and testing. These methods allowed us to not only produce the optimal integration of these technologies, but also produced qualitative and quantitative comparisons of less successful techniques so that future researchers might benefit from our extensive testing. This comparative work was performed at all levels of system development including the system architecture, control structure, knowledge representation, implementation, and error analysis.

This comparative evaluation methodology required us to design a highly modular framework in which we could prototype and evaluate the speech understanding techniques that were being developed. A hierarchical system with multi-level knowledge representations was chosen as the best approach for handling this type of comparative development. The interfaces at each level were derived from the symbolic representation of the speech used at that level of the hierarchy. The symbolic representations at each level were derived from the levels of data reduction that occurred as the speech was processed along the continuum of raw acoustic waveform to a representation of meaning. The well-defined interfaces and modular programming approach allowed head-to-head comparisons of several techniques within each level of the system hierarchy.

The system development was made possible by the use of the ESPRIT (Explorer Speech Processing at RIT) system. ESPRIT is a speech research development environment which runs on the Texas Instruments Explorer workstations. ESPRIT was developed at RIT, for use as a test-bed for the speech understanding system. The ESPIRT environment provides researchers unfamiliar with Lisp, and the Explorer workstations, the ability to develop speech and signal processing experiments. ESPRIT uses a mouse-and-menu interface combined with a graphical programming language to both design and operate a variety of speech and signal processing experiments. The work on the ESPIRT system has also led to development of an object-oriented simulation workstation at RIT Research Corporation and has involved private industry (Texas Instruments and Allied Signal).

Our research has involved researchers from many disciplines. The fields of Artificial Intelligence, Electrical Engineering, Speech Audiology and Phonology, Mathematics, and Statistics were all represented in some aspect of the project. Some of the methodologies that have been studied are Expert Systems, Neural Networks, Hidden Markov Models, Conceptual Analysis, Dynamic Programming, and Statistical Classification techniques.

Our prototype speech understanding system is a knowledge based system which attempts to capture the knowledge the experts use in reading and interpreting spectrograms. This knowledge allows us to generate phonetic information from the raw acoustic waveform. From the phonetic transcriptions, words are hypothesized and these utterances are used by the natural language system. The natural language system then analyzes the utterances and produces a representation of meaning for the utterance.

The system has been designed to be domain-independent. We have found it necessary to introduce domain-specific information at the higher level understanding functions, but this is not unusual in natural language understanding systems. The lower levels of the system hierarchy were, in fact, tested using a completely different domain than that used with the higher level understanding functions. We feel that our domain-independent approach makes the architecture of our system flexible as well as extensible.

The NAIC funded project has produced the following items:

- A prototype system with functionality and competing methods at each level of the system hierarchy.
- Four Completed Master of Science Degrees with four more pending.
- The ESPRIT speech processing system (evolved as a byproduct, it was not funded directly by the NAIC but its development was necessitated by the NAIC project. The funding was provided by RADC and Texas Instruments Inc.)
- Technology transfer from RIT to RADC as well as to private industry.

We feel that our work has effectively investigated the types of extremely difficult problems encountered when dealing with a large vocabulary, continuous speech, speaker-independent system. There are, however, some areas of research where we feel that further investigation might yield interesting results. These extensions to the work include: the development of a commercial quality speech understanding system based upon our prototype system, the incorporation of adaptive processes and learning into the system, and the testing and evaluation of as yet undiscovered speech understanding procedures.

Our work in the speech understanding area has allowed us to develop tools, technologies, and personnel that may be applied to other speech related disciplines. The speech understanding work we have done has extensions in the areas of: speaker identification, language identification, and key word spotting.

8.2 Research Project Definition

In order to effectively summarize the work of RIT over the past five years it is necessary to examine the original goals and objectives of our work. This discussion will establish the context of our research to more clearly show where we started, where we are today, and the evolution of the project over its five year life span.

8.2.1 RIT's Project Mission

As expressed in our proposal to RADC [RITR84], RIT's mission in the NAIC is twofold. Our primary research goal was the application of Artificial Intelligence techniques toward the development of speech understanding systems. More specifically, our research was geared toward a speaker independent, continuous speech, large vocabulary type of system. These types of systems are the most challenging systems to develop, but they provide the most natural interface between man and machine.

Our secondary goal was to support and implement the *mission objectives* of the NAIC here at RIT. These objectives being: (1) AI technology advancement needed to support knowledge-based systems applications to C3I mission requirements; (2) The advancement of the RADC in-house research and development capability; and (3) Education and training in AI technology to expand the quantity of AI researchers and faculty.

8.2.2 Project Specifications

In order to pursue the goals above, overall project specifications were developed which gave the project both scope and direction. These specifications helped to keep the various sub-research projects oriented toward the higher level mission objective as well as producing a cohesive research project after such a lengthy investigation.

The specifications for the Speech Understanding System were as follows:

- Derivation of an intermediate phonetic representation for continuous speech from any speaker

- Measure the quality of the match between errorful phonetic representations and phonetically based lexical entries

- Incorporate multiple level knowledge sources to differentiate plausible and implausible parsings.

- Extensively research, test and compare competitive methods used in the system

Ancillary goals for the support of the NAIC were:

- Form a core group of faculty with education, experience and interests in AI

- Enlarge and strengthen the computer science graduate program in the area of AI

- Increase AI knowledge of local industry.

- Obtain hardware and software tools necessary to do AI research at RIT.

8.2.3 Speech Understanding Project Focus

The application of AI techniques to the problem of developing Speech Understanding Systems, has given our work a unique flavor and focus. This focus is based on modeling the human ability to understand speech. This modeling occurs at many levels. For example, we may model the human auditory system's ability to classify phonetic categories, or we might model the human ability to transcribe utterances based upon their spectrograms. These models, as well as others, help us to identify the knowledge sources and representations that are applied to understanding speech. Like many problems in AI, we are attempting to investigate a system of processes that is poorly understood, difficult to analyze/dissect/measure, and performed almost effortlessly by human beings.

Thus our approach to understanding speech is based on techniques which, at some level, model the human ability. This approach fundamentally differs from the acoustically based engineering techniques that were once applied to speech recognition problems. This approach does not imply that the structure of our system is indicative of the methods used by humans, but instead tries to build on the processes that are demonstrated by humans. Consider the case of stop consonant place of articulation. If experimental results indicate that humans can classify place of articulation to high degree of certainty it is reasonable to try to design a system which also models this behavior. This is true even if it is not understood how this information is used in the understanding process. This approach is exemplified in our use of many feature extractors. These feature extractors attempt to extract the same types of information which speech scientists believe are captured by the ear. Thus, we attempt to capture the knowledge produced by the physiology of the ear.

8.3 Speech Understanding Project Design

Over the course of five years it is not unreasonable to see some evolution in the system design of a project. In our situation, design changes were generally either the result of new hardware architecture, or flaws in our previous design which became apparent as we attempted to implement it. The overall system goals remained constant throughout and these goals are reflected in the software architecture of the system. The system architecture has both hardware and software components. The hardware architecture primarily involves the machine platforms and speech processing hardware upon which the system was implemented. The software architecture involves the knowledge representations and decision mechanisms necessary for understanding speech.

8.3.1 Speech Understanding Project Software Architecture

The software architecture has always been viewed as a knowledge based system that attempts to capture spectral information from a spoken utterance. Experts who read and interpret spectrograms use this information to formulate hypothesis about the unknown utterances. Combined with information observed directly from the audio waveform and a rich set of knowledge sources, experts can accurately segment and parse unknown utterances.

The system architecture was designed to capture the low-level information present in the audio signal and transform it into knowledge representations which can be used to parse the utterance. This transformation from audio waveform to low-level knowledge representation to ever higher-level knowledge representations is both a means of data compression as well as a method to reduce the complexity of the overall problem. The data compression occurs because the representation of the signal at each level in the hierarchy incorporates a greater amount of knowledge than the level below it.

The lowest level in the software architecture is the audio waveform. This waveform is digitized and analyzed using standard speech and signal processing algorithms to obtain low-level features of the signal over time. These algorithms include FFT and LPC analyses, formant and pitch tracking, zero-crossing counts, etc. These features capture both the spectral and intensity information that is processed by the ear.

So at this level we have transformed the analog, acoustic information from an utterance into a set of discrete vectors, where each vector represents the feature values for a short interval of the utterance. The reduction of data at this level is substantial and results in a speech representation of the utterance that does not sacrifice information necessary for higher-level decision making processes.

The sequence of feature vectors is then analyzed by a classifier that makes decisions about coarse phonetic categories. The classifier segments the signal into discrete segments based on the categories: vowel-like, strong fricative, weak fricative, and silence. These segments can be thought of as regions of the signal that are roughly homogeneous. This sequence of coarse phonetic categories is the next speech representation in the software architecture hierarchy.

These sequences of coarse phonetic categories are presented to classifiers that attempt to assign phonetic labels to the coarse phonetic segments, ie. the coarse phonetic segments are themselves segmented into actual phonemes. This classifiers do not necessarily identify a single phoneme for each segment, but have the ability to generate probability measures for several possible phonetic labellings. This approach of assigning confidence factors to the labellings mirrors the approach of human spectrogram readers.

This lattice of phonetic labels and probabilities is presented to a word hypothesizer. It is the hypothesizer's responsibility to generate possible word sequences from the strings of phonetic labels. The hypothesizer must address the following problem areas: (1) phonetic insertions and deletions due to the context of the word; (2) efficiently searching the lattice of phonetic labels and probabilities for the correct combination; (3) handling errors made by the lower-level classifiers.

Once we have the candidate word sequences, they are analyzed by a module which attempts to select the best word sequence based on confidence factors from the lower-levels of the system and on domain-specific knowledge. This selected utterance is considered to be the correct transcription of the raw signal and is passed on to the natural language understanding system.

The highest level of the system is a natural language understanding system. This system builds a representation of meaning for the input utterance using all possible knowledge sources including domain knowledge, syntactic and semantic information, and domain goals.

The software architecture is best described as a data driven or forward chaining type of control strategy. The strategy is based on the assumption that a reasonably accurate phonetic transcription of the raw speech signal can be produced by the low level modules in the system. We are interested in phonetic transcriptions because they are reasonably speaker independent within the constraints of nationality, region, language, and context.

8.3.2 Speech Understanding Project Hardware Architecture

The hardware side of our speech understanding work has changed greatly over the course of the contract. The system implementation began in 1985 on a Sun Microsystems model 2/130. This platform had been chosen because of the availability of speech analysis software for the Sun 2. This speech analysis software was proprietary and not available for distribution or resale. It was developed by Speech Recognition Systems, Inc of Rochester, New York for their own commercial applications. It was provided to us, at no cost, through our close association with Dr. Robert Houde of Speech Recognition Systems with the understanding that it will be used only for the research of the NAIC speech understanding project. This analysis software allowed us to perform many signals analysis functions such as: A/D and D/A conversion; display of 2 dimensional analysis including waveforms, zero-crossings, energey measures, etc., FFT and LPC analysis; and spectrogram displays. The Sun became our main speech research platform. We extended our analysis capabilities by developing new analysis programs including a formant tracing tool based on LPC coefficients, formant and pitch tracking software, and speech synthesis software. The Sun also held the Carnegie-Mellon University (CMU) speech data base which we have used throughout the development of the speech understanding system. The CMU database consists of approximately 1300 utterances broken down into vowel-dense, fricative-dense, and stop-dense categories. For each utterance CMU provided us with the digitized speech and a hand-labeled phonetic transcription of the utterance.

As the project progressed, the Sun became inadequate in terms of computing power and storage. While we were considering upgrading the Sun 2 to a model 3 the NAIC announced the availability of university priced Texas Instruments Explorer I LISP workstations. RIT acquired two of these workstations in the spring of 1987. These workstations provided us with a better platform for implementing the higher-level decision algorithms in the project as well as the lower-level feature extraction. Odyssey TMS320 Signal Processing boards were acquired and dramatically improved our signal processing capabilities. The Explorer I workstations were later upgraded to Explorer II workstations and increased system performance even more.

A speech analysis workstation similar to the SRS analysis software on the SUN was needed for the Explorer platforms. This need prompted the development of the ESPRIT speech workstation. Funded jointly by TI and RADC, ESPRIT met and exceeded the functionality available on the Sun. The ESPRIT system has evolved from a simple signal analysis tool into an integrated, speech-research environment. As ESPRIT is the primary delivery vehicle for the speech understanding system, it will be discussed at some length in the following section.

8.3.3 The ESPRIT System

ESPRIT is a speech research development environment [RITR89] that runs on the Texas Instruments Explorer LISP workstation, optionally augmented with one or more Texas Instruments Odyssey Signal Processing boards.

ESPRIT's main goal is to provide speech scientists, linguists and engineers with an intuitive software environment in which to study speech signals and to provide tools for conducting speech research. The basic functions of ESPRIT are to collect, process and graphically display raw and processed speech signals in ways that are useful to speech scientists. No prior knowledge of LISP or any other programming language is necessary, and no prior knowledge of the operation of the TI Explorer is required in order to perform a wide variety of speech processing tasks.

Users may operate ESPRIT interactively to perform simple operations one at a time and display the results after each operation is performed. These operations and displays include raw waveforms, FFT and LPC spectrograms, and other useful parameters and features that can be extracted from speech signals.

Users may also build modules made up of simpler operations and displays to perform complex tasks. This feature allows users to literally "draw" a sequence of speech processing functions and display directives and then execute the resulting "program" to perform the task that was drawn. This allows a user who is not a programmer to put together existing programs into a configuration that performs some desired task without having to type a single line of code.

The ESPRIT user interface takes a mouse-and-menu approach, and in fact, the entire system can be run by clicking the three buttons on the Explorer's mouse. Help is available at all times for all commands, both in the form of mouse command documentation, which is always displayed automatically, and in the form of more extensive documentation, which may be displayed easily on demand.

Care was taken in the design of ESPRIT to make the displays and mouse buttons as consistent as possible. This helps the user to develop sound instincts for how to use the system and view the displays. For users who feel a need to type keystrokes instead of navigating through menus, all commands on all menus have corresponding keystroke equivalents.

Both the module building capabilities and graphical display capabilities were heavily used in implementing the speech understanding project on the Explorers. The section of ESPRIT used for building modules is the Module Editor. Through the Module Editor users can build up complex processes by describing how the pieces fit together graphically. Figure 8-1 on the next page shows the software architecture of the speech understanding system as it is built under ESPRIT. Figure 8-2 shows some of the graphs that ESPRIT can generate. These graphs allow us to evaluate the signal processing algorithms used in the system and to make exact measurements from the analyses.

Figure 8-1 Meta-Module of Speech Understanding

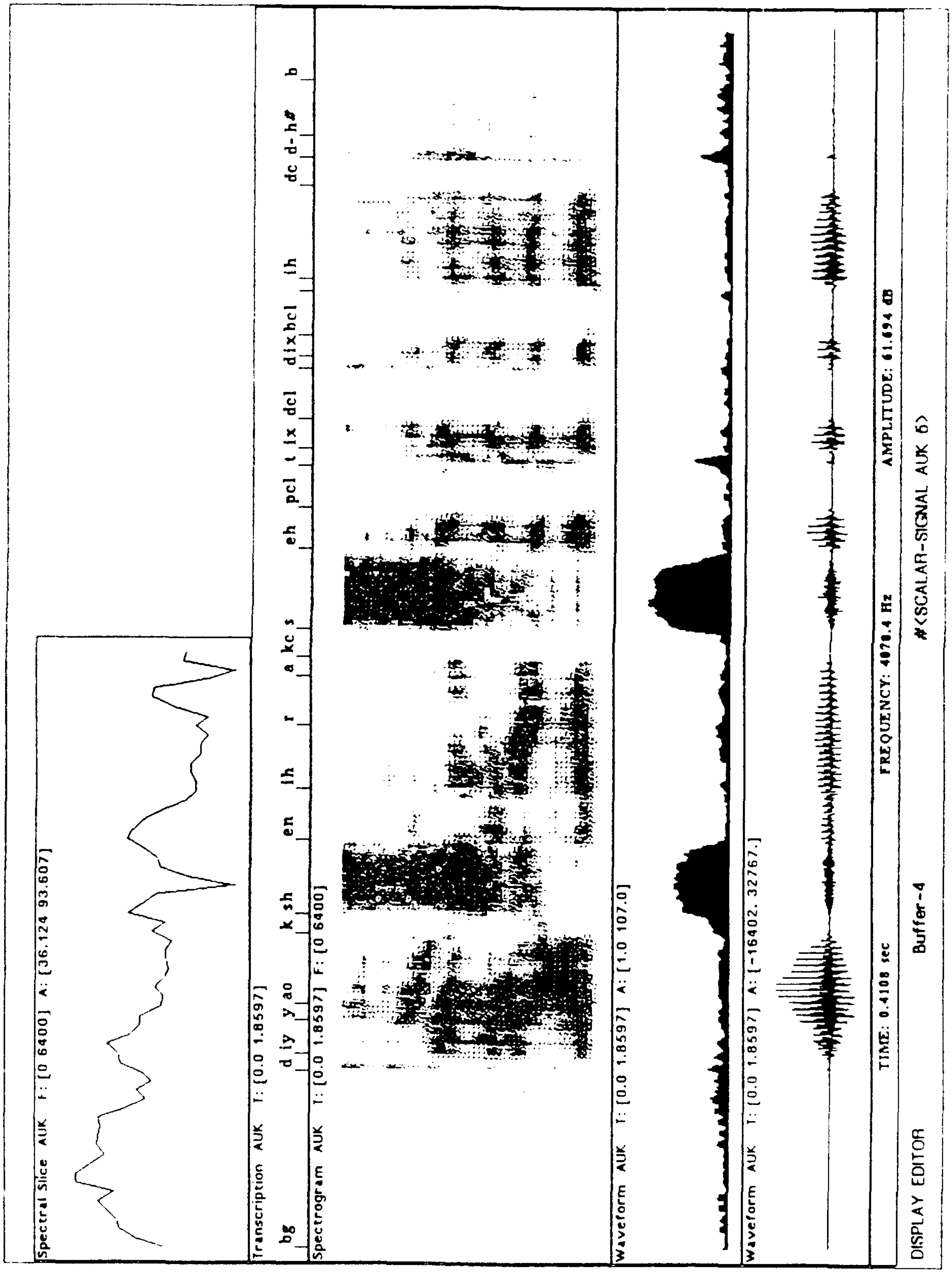
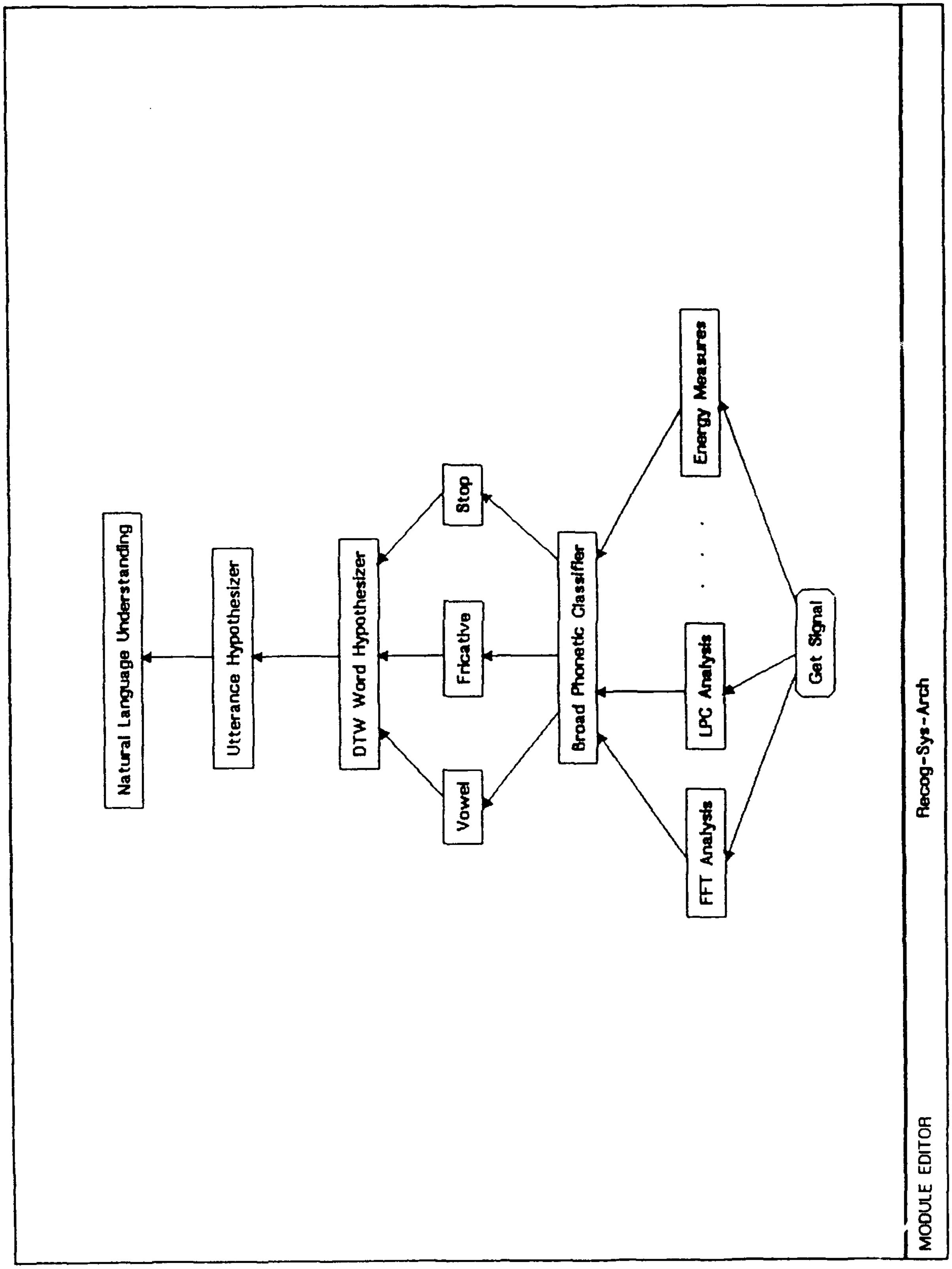

Spectral Slice AUK F: [0 6400] A: [36.124 93.607]
Transcription AUK T: [0.0 1.8597]
bg d iy y ao k sh en ih r a kc s eh pcl t ix dcl dix bcl ih dc d-h# b
Spectrogram AUK T: [0.0 1.8597] F: [0 6400]
Waveform AUK T: [0.0 1.8597] A: [1.0 107.0]
Waveform AUK T: [0.0 1.8597] A: [-16402. 32767.]
TIME: 0.4108 sec
FREQUENCY: 4070.4 Hz
AMPLITUDE: 61.694 dB
DISPLAY EDITOR
Buffer-4
#<SCALAR-SIGNAL AUK 5>

Figure 8-2 ESPRIT's Graphical Capabilities



The ESPRIT environment itself, is an object-oriented system built around the following conceptual objects: data objects, processes, displays and meta-modules. Data objects are data structures which hold raw speech, sequences of FFTs or LPC spectra, sequences of phonemes or words. These data objects may be permanently stored as files or dynamically created and destroyed throughout the execution of the user's application.

Processes are TMS 32020 code or LISP code which are used to create, analyze, and destroy the various types of data objects. ESPRIT contains a large number of signal processing routines that may be used by any application. Users may also develop their own processes and incorporate them into the ESPIRT environment.

Displays are the graphical windows which are used to display and measure the data objects. Several types of displays are seen in Figure 8-2. The environment stores the knowledge to correctly display the various types of data objects, or the user can specify a different type of display other than the default.

The most important capability provided by ESPRIT is the ability of users to build their own meta-modules. A meta-module is a directed graph that contains process objects, display objects and possibly other meta-modules as well. A graphical interface allows the users to draw their applications. Figure 8-1 is a meta-module which describes the speech understanding system.

ESPRIT has given RIT the capabilities to not only study speech, but to study many different areas of intelligent signal processing. The ESPRIT environment could be used to study image and vision problems, sensor fusion and radar signals, language and speaker identification, and key word spotting. ESPRIT's non-programmer-specific interface has also been the basis for the development of an object-oriented simulation package for the Explorer systems. This simulation environment is currently being used to study manufacturing simulations, and distributed discrete event simulation (DDES).

8.4 RIT's Speech Understanding Methodology.

As indicated by the software architecture, the understanding system can be broken down into three sub-processes: (1) The transformation of the audio signal into a speaker-independent phonetic transcription; (2) The transformation of the phonetic transcription into the best representation of the utterance at the word level; and (3) determination of the utterance's meanings based on the words. This section will investigate the methods that were employed to complete these sub-processes.

8.4.1 Low Level Feature Extraction

The reasoning behind the investigation of low level features is that they follow the model of the human auditory system. The auditory system is capable of very accurately measuring both time and frequency events. Thus the features we have investigated have been of two classes, short time analyses of energy information and analyses of frequency information. Each of the low level features we have investigated, and the reason behind its inclusion in our research, will now be briefly discussed.

Energy measurements can be computed to measure the sound pressure of the signal over some number of samples. These energy measures can be used for silence detection, marking prosodic features. Relative energy measures (measures which relate energy at certain frequencies) can reveal valuable information for classifying certain similar phonetic categories. Relative peak energy measures examine the relationship between total energy for the samples and the peak value for the samples. Relative peak energy is generally used when a normalized energy measure is required.

Zero crossing rates and counts are critical in the detection and classification of fricatives. This feature measures the number of times the signal crosses the zero-amplitude measure. A dead band is usually implemented to reduce the impact of low-level background noise. Sound pressure values within this dead-band are not counted.

FFT spectra are calculated by applying a Fourier analysis to the time-order signal to produce amplitude/frequency pairs over time from the raw waveform. The size of analysis window controls whether wide-band or narrow band frequency analysis results. FFT Spectra are the basis for relative spectral energy measures, spectrogram displays, and most other measures which require frequency information. For speech signals, LPC Spectra are often used instead of FFTs.

LPC Spectra are produced by applying Linear Predictive Coding algorithms to the speech signal. LPC analysis models the speech vocal tract as an all-pole filter whose parameters can then be used to replicate the original speech waveform. LPC spectra tend to show better resolution of formant frequencies (resonances produced by the vocal tract) than FFT spectra. The order of the LPC filter controls the number of spectral peaks sought by the model. A fourteenth order model will find 7 peaks which is more than adequate for most applications.

Average spectra are computed to smooth the spectral change over time. This is most often done in order to better display the spectra in a spectrogram or waterfall display. It is not generally used as a feature for higher level decision making.

Spectral moments reveal information about the distribution of energy across the frequency range. The first four moments indicate mean, variance, skewness, and kurtosis. Mean is midpoint or average frequency of the power distribution. Variance indicates how compresssed or spread out the energy is across the frequency range. Skewness is a measure of how symmetrically the energy is distributed about the mean. Kurtosis measures the amount of energy at the extremes of the spectra relative to the amount of energy in the center. These four spectral moments have been used in the analysis of stop consonants and vowel classification.

Some investigation has gone into using the analysis of frequency versus energy measures. These calculations are made against the frequency information of the spectra as opposed to the four spectral moments which are computed from the energy information. These features are being used by the vowel and stop consonant classfiers.

Formant traces have been calculated using a variety of methods. The ESPRIT system currently implements the Markel [MARK76] algorithm which examines spectral peaks in the LPC spectra. Other methods which have been examined use zero-crossings, spectral moments, and vector quantization symbols. The greatest amount of research in this area was the development of some statistical approaches to formant tracking [GAYV89]. This approach assigns probability measures to sets of features extracted from a short-time analysis of the signal and a conditional mean estimate is used to determine formant frequency values. This work is a generalization of methods introduced by Kopec [KOPE86] based on hidden Markov models [RABI86] and vector quantization symbols. The results of this research will be covered in the results section of this report. Formant frequency information is used extensively in vowel and vowel-like classification.

Pitch traces have been studied using a variety of methods. Most of the methods are based on an autocorrelation method. This method shifts a window of the input signal along the signal and computes the corresponding autocorrelation function. The maxima of this function represents the area of highest correlation and corresponds to a shift of one pitch period location. The fundamental frequency is then calculated from the pitch period. Another pitch tracking method is Markel's Simplified Inverse Filter Tracking algorithm (SIFT) [MARK72] . This algorithm combines standard autocorrelation techniques with inverse filter formulation and cepstral information. The pitch period of this glottal waveform is then estimated using an autocorrelation technique. Pitch trace information can be helpful in normalizing frequency information to achieve independence between male and female speakers.

In general, sets of features are collected over some window size of the signal. These feature vectors are then collected into a knowledge representation known as a pattern set which can be used by the higher level classifiers.

8.4.2 Coarse Phonetic Classification

Our studies in coarse phonetic classification of speech signals [DELM88] attempted to answer two questions. The first question is whether or not our low level features preserve the information necessary to perform coarse phonetic classification. If our feature sets are not strong enough to indicate coarse phonetic categories we would expect poor performance from fine phonetic classifiers using the same information. The second question that needed answering was whether it was better to separate the segmentation and identification problems. In other words, does the system achieve better classification accuracy with a coarse phonetic classifier doing segmentation and then identifying the phonemes within each segment, or is it better to determine phonemes directly from the feature vectors without first segmenting the signal. To answer these questions we had to implement a coarse phonetic classifier.

The coarse phonetic categories studied were: vowel-like, strong fricative, weak fricative and silence. Two decision making procedures were investigated. The first procedure used a Euclidean distance measure to clusters in a n-feature dimensional space that had been derived using the K-means clustering algorithm. The second procedure investigated the use of a multivariate maximum likelihood distance measure to classify the segments. Also investigated was the impact of structuring the decision making process. Several tree-structured decision architectures were compared for each of the classification methods.

Both methods were trained on ten speakers using 98 utterances from the CMU speech database. Training and testing involved both known and unknown speakers. The results of this study are presented in section 8.5.2 of this report.

The coarse classifier must be able to handle the problems of speaker variability as well as the coarticulatory affects produced in continuous speech. Any gross mistakes in segmentation or classification (e.g. identifying a vowel as a fricative) will probably result in an ultimately incorrect word identification.

The architectures of both decision methods begin with the generation of label/vector pairs (LVPs) for each 10ms frame in the utterance. This LVP consists of a phonetic label, which is taken from hand labeled information provided by CMU (ie. a priori information) and a feature vector computed for that frame. The features used for this project were: zero crossing rate, total energy, relative energy, peak energy, measure of spectral change, measure of periodicity, and the four spectral moments.

The training of each of these classifiers is somewhat different from this point. The maximum likelihood classifier trained using a supervised algorithm. LVP's from the same coarse phonetic class are clustered together. For each of the four clusters, a mean feature vector and inverse covariance matrix are calculated. These statistics indicate the cluster's center and distribution in ten dimensional space (from the ten features in the vector).

The training used by the K-means method is unsupervised and uses no a priori information about the actual phonetic label for each LVP. The K-means algorithm is a clustering technique that will form K clusters of data points in an n-dimensional space. For speech samples the feature vector portion of the LVP is considered to be a point in an n-dimensional space. The algorithm initially chooses K initial centers (often the first K data points) and distributes the remaining data points about the cluster centers based on a Euclidian distance to the cluster centers. New centers are calculated by choosing the point such that the sum of the squared distances from all the points within the cluster is minimized. This distribution/recalculation of points and centers continues until none of the cluster centers are different from the previous calculation. Note that this process does not necessarily cluster data into the coarse phonetic classes. It clusters based on similarities in the feature vectors for the data points. If the feature vectors do not indicate a reasonably separable n-dimensional space then the cluster analysis will not yield the coarse phonetic categories we are interested.

After both methods have been trained, the classification of a LVP from a test utterance is quite similar. A distance measure is calculated between the unknown LVP and cluster centers that have been calculated by the K-means and maximum likelihood training. A z-score is used as the distance measure when using the K-means centers. The maximum likelihood method uses the following distance measure:

$$r^2 = (x - u)^t \ \mathrm{SUM}^{-1} \ (x - u)$$

where:

x = the feature vector being evaluated
u = the mean feature vector for the cluster
SUM^{-1} = the inverse covariance matrix for the cluster
$(x - u)^t$ = the transpose of $x - u$

The following possible decision tree structures where used:

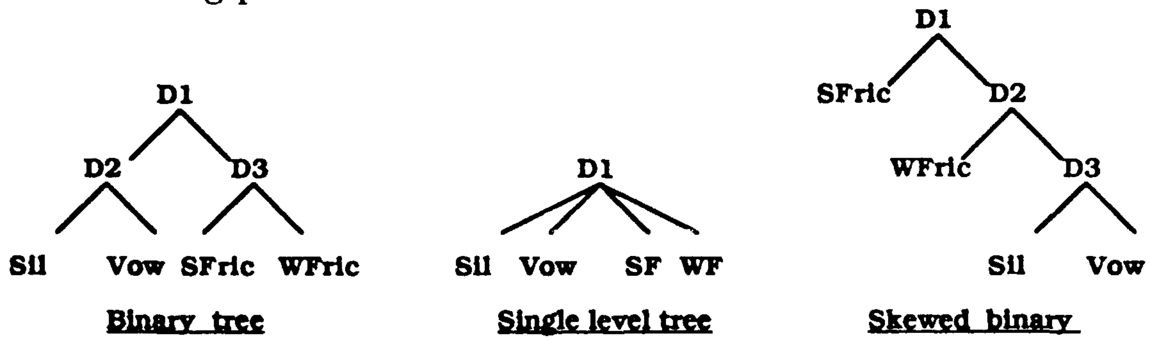


Figure 8-3 Three decision tree structures for coarse classification

In figure 8-3 the nodes labeled with a D represent a decision point between the children. These indicate the optimal combination of classes within the three types of decision tree structures. In the skewed tree the goal is to find the most identifiable class first, then the next easiest, and so on. The tree structure then affects the classes that you train the classifier to detect. With the binary tree the K-mean training algorithm would generate two clusters and then split the clusters again.

8.4.3 Fine Phonetic Classification

Once coarse segmentation and classification has been performed, the system must attempt to produce the actual phonetic transcription of the utterance. This is done using several fine level classifiers. We have done research in the areas of fricative, stop consonant, and vowel identification. Each of these classifiers takes the appropriately labeled segment that was indicated by the coarse level processes as well as the features for that segment and attempts to identify the phoneme(s) present in the segment.

8.4.3.1 Fricative Expert System

In our investigation of fricatives we examined the use of an expert system to classify fricatives based on expert spectrogram readers [ATKI87]. Two experts and a knowledge engineer were used to build the expert system using Rulemaster, a rule-based expert system shell from Radian Corporation.

Using Rulemaster, the knowledge engineer produced example tables indicating the conditions necessary for certain actions to occur. From these example tables Rulemaster induced the rules necessary to build the decision tree. This method requires that knowledge of all possible values of the conditions or attribute be known at system development time.

The main example table for the fricative expert system appears below. Fricatives can be classified based on the attributes of voicing and place of articulation. Voicing is present when the vocal cords are vibrating and is indicated in spectrograms by the presence of a voice bar. The voice bar is the frequency resonance that appears in a spectrogram as a single dark bar at the fundamental frequency of the speaker. Place of articulation is the location of the articulatory mechanism which produce the sound. The locations used by this system were labiodental, alveolar, palatal, and dental, which indicate mechanisms at the lips and teeth, alveolar ridge, palate, and teeth, respectively.

Place	Voicing	Fricative Indicated
labiodental	absent	*f* as in *foo*
labiodental	present	*v* as in *vote*
alveolar	absent	*s* as in *see*
alveolar	present	*z* as in *zoo*
palatal	absent	*sh* as in *shoe*
palatal	present	*zh* as in *azure*
dental	absent	*th* as in *thief*
dental	present	*dh* as in *then*
uncertain	present	*v* or *dh*
uncertain	absent	*f* or *th*

Table 8-1 Example table for classifying fricatives.

The high level goals of place and voicing are then proved with rules induced from example tables for the place of articulation classes and voicing attributes. The classification results and a sample expert system interaction are presented in section 8.5.3 of this report.

8.4.3.2 Stop Consonant Classification

Our work in stop consonant classification [CAMP89] is based on the shape of spectra taken from the burst region of the stop. Acoustically, stop consonants appear as a short period of low signal energy (closure) followed by an abrupt release (burst). This release appears as a large body of noise in the spectra of the stop. The durations of the bursts are relatively short, typically 20 to 40 msec. It has been shown that recognition of stops can be achieved knowing the shape characteristics of the stop spectra [STEV78, BLUM79, KEWL83]. Spectral moments have been used as features to capture the concepts of spectral tilt and compactness.

Our general approach to stop classification has been the following: (1) Spectral Analysis of the stop is performed; (2) Feature Extraction from the spectra; (3) Optional feature compression; (4) Maximum likelihood classification.

Several signal analysis methods were compared to obtain both static and running spectra for the stop. Data analysis windows varied from 10, 15, 20, 30 and 40 msec. The windowing function used was a cosine modified Hamming window which preserved data on the left side of the window. The number of spectra analyzed varied from one to six with a shift size equal to 50% of the data window. Both FFT and LPC spectra were investigated. For the static spectra 256 and 1024 point FFTs were examined. In all combinations the log power spectra was the basis for feature extraction.

The features that were used were the moments of a distribution, mean, variance, skewness, and kurtosis. The central clustering of a distribution is usually measured by the first moment (the mean). In a normalized distribution, mean is only relevant for values along the abscissa (frequency values). Three estimators of central clustering were examined: Mean of the abscissa (Center frequency); Mean of the ordinate (meaningless for a normalized distribution); Median (value for which large and smaller values of x are equally probable). Two estimators of dispersion were examined: The variance and the mean absolute deviation. Skewness measured the degree of asymmetry (tilt) of the distribution. Kurtosis measured the peakedness or flatness (compactness) of the distribution.

The classifier being used is the maximum likelihood classifier that was discussed in the coarse phonetic classification section above as well as the vowel classification section that follows.

The stop data was separated into several classes to enable a reasonable comparison of classification results. The following figure shows the statistical breakdown that will be used in the results section.

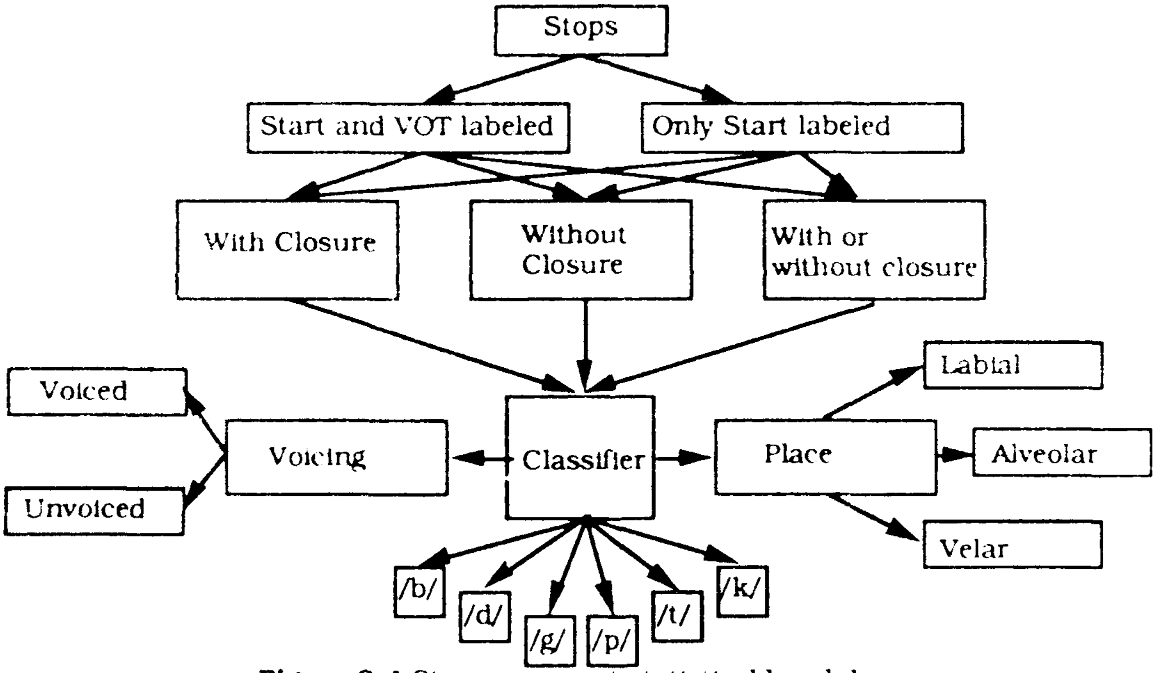


Figure 8-4 Stop consonant statistical breakdown

Like fricatives, stop consonants can be classified by their place of articulation and voicing characteristics, hence the testing of the classifiers ability to predict place and voicing. The separation of stops with closure and without closure is to determine the affect of not having the silence from the closure as an identifying feature of the stop. There has been little work in classifying stops where the closure segment is not clearly present. The separation of stops with VOT (Voice Onset Time) marked is to determine if knowledge about the VOT is useful in classifying the stop. The classification of stop consonant results are presented in section 8.5.4.

8.4.3.3 Vowel Classification

The performance of two types of classifier have been investigated for use with vowel and vowel-like segments. The first uses a maximum likelihood approach similar to the one used in the coarse phonetic segmentation. The second type uses a back-propagation neural network. These classifiers produce a phonetic decision for each 5ms frame of the analyzed vowel-like segment. This output sequence of phonemes is then presented to a Hidden Markov Model (HMM) which produces the final output string. These two approaches to coarse phonetic segmentation might better be called pre-classifiers as they attempt to reduce the complexity of the classification performed by the HMM.

The reason for using the HMM as the final decision process was to make use of the temporal information in the signal. One of the keys to vowel segmentation is identifying the formant transitions within the vowel. The HMM has the ability to capture these temporal characteristics whereas a static classifier cannot. The reason for using the neural net and maximum likelihood processes at all is to reduce the time involved in running the HMM. It is computationally prohibitive to run every feature through the HMM.

Like all of the fine level sementers mentioned so far, the vowel classifier makes use of low level features from the signal. The features used by this classifier were the four spectral moments, fundamental frequency (SIFT algorithm), and formant traces (using hand labeled traces from LPC spectrogram).

The vowels being examined are the ten vowels used in the classic Peterson and Barney vowel perceptions study [PETE52]. They are /iy, ih, eh, ae, o, ah, uw, aa, er/.

The maximum likelihood pre-classifier is nearly identical to the maximum likelihood process described in the section on broad phonetic segmentation and will not be discussed in detail.

The neural network pre-classifier uses the back-propagation training algorithm. This algorithm is based on a multi-layer feed-forward perceptron. We used a single hidden layer with the hyperbolic tangent as its nonlinearity function. Since several feature set combinations were tested, and the number of hidden nodes is dependent on the number of input values, several different number of hidden layer nodes were investigated.

The decision to use a HMM was based on its ability to accurately model a time varying process. The HMM does this by assuming that a time-varying process can be thought of as a set of states with transitions between the states. Through presentations of example processes the HMM can be trained to predict the probabilities of the transitions from state to state and the probabilities of an output symbol being generated by each state. In applications where HMMs are applied the transitions from state to state are not directly observable. In the vowel study the transitions being modeled are primarily the transitions in the formant frequencies observed in a vowel-like segment. These formant transitions are produced by the time varying process of articulating continucus speech.

A three state, left-to-right model was used as the HMM for this study. In this model each of the three states has a possible transition to itself or to the state immediately to its right. The rightmost state has a transition to the leftmost state. Three states were chosen to attempt to model the onglide, central, and offglide segments in a vowel's formant frequency transition. The vowel classification results are presented in the section 8.5.5 of this report.

8.4.5 Automatic Neural Networks

RIT has become very interested in the application of Automatic Neural Networks (NNs) to solve classification problems. Neural networks are not difficult to program and quite simple to use. All they require is a set of features as input and they produce a sequence of discrete classification symbols. There is a natural inherent parallelism in neural networks which can be exploited in terms of fault-tolerance and execution speed.

The ability of NNs to learn by example allows them to be easily trained. The more training data that is presented to the network, the more accurate the results. NNs can be taught to "remember" certain classes of objects. The features of the objects can be presented at the input layer of the network and the corresponding class at the output layer of the network. The internal nodes of nets can then be trained to adjust themselves to produce the desired output. This is an example of supervised training, the person training the net knows the correct response and *penalizes* incorrect responses. The penalization is done by modifying the behavior of the internal nodes in the network. After some number of trials the NN can consistently produce the desired output.

Unsupervised training is used when the desired output symbol is unknown. The network forms internal clusters which it uses to classify the data into the discrete symbols at the output layer.

Neural networks can be applied to many classification problems. We have utilized them primarily for speech-related classifications such as coarse phonetic classification, Phonetic identification and classification, and word recognition. NNs in general can be applied to any feature-based classification problem. NNs can address the same class of problems that have been traditionally solved with statistically based classifiers such as K-means cluster analysis and maximum likelihood.

Figure 8-5 shows the structure of the networks that we have used at RIT. This network has three nodes in the input layer (X0, X1, X2), five nodes in the hidden layer, and nine nodes in the output layer. This net would be used to classify data into nine discrete classes. We have used twice the number of input layer nodes minus one as a formula for computing the number of hidden nodes in a single hidden layer model. For this figure the three input layer nodes correspond to three features that would be used for the classification. The networks we have investigated are fully connected, with each node in one layer attached to each node in the level below. Weights are attached to each connection and are adjusted by the training algorithm to produce the desired values at the output layer when presented with certain values at the input layer.

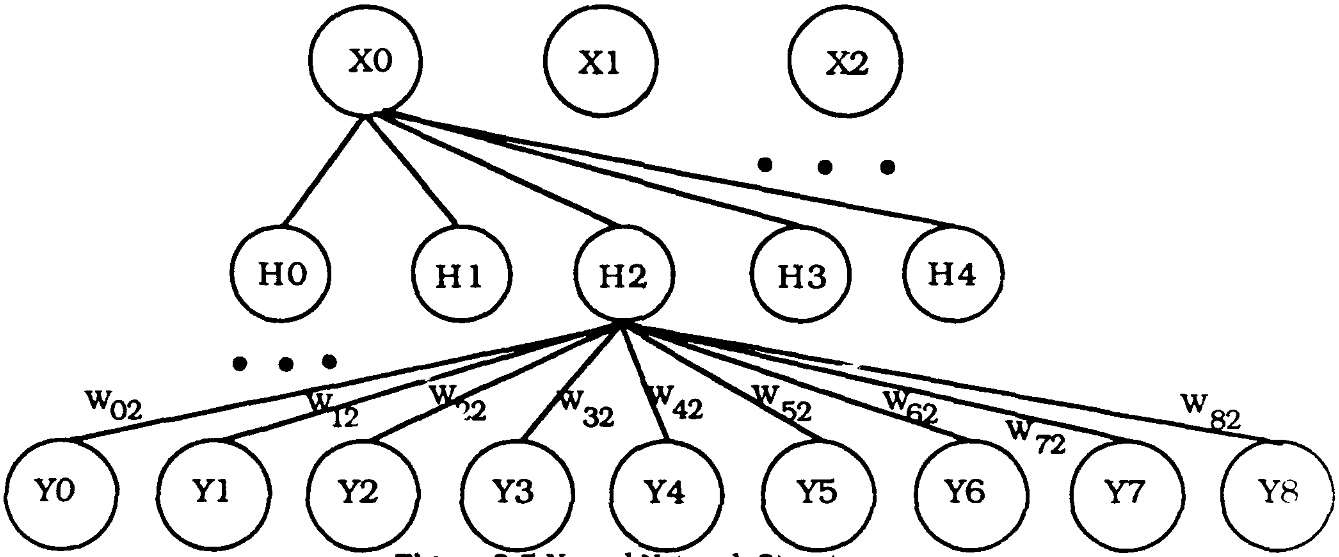


Figure 8-5 Neural Network Structure

We have been using a supervised training model called back-propagation. In this model input is presented at the input layer and if the output layer is correct then no adjustments are made to the weights in the system so as to correct the error. The weights are adjusted according to the equation:

$$W_{ij}(t+1) = W_{ij}(t) + \varepsilon \delta_j x_i + \alpha(W_{ij}(t) - W_{ij}(t-1))$$

where

ε = Gain term

α = Momentum term

$W_{ij}(t)$ is weight from hidden node i or from an input to node j

x_i is either an input node or a node in the hidden layer

Calculation of δ_j based on derivative of the hyperbolic tangent

$\delta_j = (1-y_j^2)(d_j-y_j)$ when j is a node in the output layer

$\delta_j = (1-x_j^2)\Sigma(\delta j-W_{jk})$ when j is a node in a hidden layer

d_j is the desired output for the specific node j and

y_j is the actual output for node j

When using the back-propagation model the initial weights are randomly set to small values (≈ 0.05). The training is simple, but computationally expensive. For very large feature sets the back-propagation training method can be very slow. For example, on an Explorer II, a 10,000 token feature set, presented to a neural net with 10 input layer nodes, 19 hidden layer nodes, 10 output layer nodes, took twelve hours to train to 50 passes (presentations of the input). There is quite a bit of interest in moving some neural network software to digital signal processing boards such as the Odyssey board for the TI platform or the DSP32 board we have used in the Macintosh environment. Transputer boards may also be used for high-speed training of neural networks.

Our interest in neural networks helped to develop an informal weekly neural network seminar at RIT. At this seminar other types of networks and training models were examined. RIT also sponsored the 1989 NAIC Spring Meeting on Neural Networks and Complex Distributed Systems. At this meeting several NN topics were discussed including associative memory of Hopfield networks, Neural Networks for handwritten digit recognition, Neural Networks for phoneme classification, Electronic Neural Networks, and others.

8.4.6 Word Hypothesis from Errorful Phonetic Strings

This is the first project which operates on the phonetic strings produced by the lower levels of the system. The phonetic string representation of the speech signal is intended to be speaker independent. This project [SELL89] investigated a dynamic programming approach to the word hypothesis problem. It was based on an approach known as Dynamic Time Warping (DTW) [ITAK75]. DTW is a common method of sequence comparison used in matching a reference vector with an unknown vector. As applied to phonetic strings, the DTW algorithm compared the unknown phonetic string with reference strings from a database of words. A cumulative least cost path combined with an empirically determined threshold was used as the decision criteria for recognition.

This work is the first of the higher-level components of the speech understanding system. The knowledge representations used at this level are closer to the meaning of the speech than those at the lower levels of the system. Both this work and the later natural language understanding work are working in the domain of cockpit-speech. The CMU speech database is not oriented toward any domain (ie. the utterances were chosen for their acoustic properties not how they apply to any type of domain or scenario). Thus the higher-levels in the speech understanding system investigated the speech used by fighter aircraft pilots. The utterances were taken from a USAF Cockpit Natural Language study [LIZZ87]. The vocabulary from the study involved a vocabulary of 656 words gathered during simulated aircraft missions.

The difficulties in matching the unknown phonetic strings with the reference string are the result of two classes of problems . The first problem is the front-end errors produced by the lower-levels of the speech understanding system. These front-end errors are due to the inability of the lower-levels of the system to differentiate similar sounding phonemes. The second class of errors consists of errors produced by the speakers. This second class of errors consists of insertion, deletion, and substitution errors. These errors are more common in continuous speech where we are less precise in our pronunciation of words. The following are examples of the three types of errors.

Insertion Error — *chauffeur* /sh ow f r/ is pronounced /sh ow **l** f r/
Deletion Error — *hallway* /h ao **l** w ey/ is pronounced /h ao w ey/
Substitution Error — *tell* /t eh l/ is pronounced /**k** eh l/

Some of these pronunciations are rule-governed. For example, the word identify is often pronounced idenify. The rule governing this deletion states that the phoneme /t. may be deleted when it appears in between /n/ and a vowel. These phonological variations can be handled by creating an an idealized pronunciation in the lexicon and alternative pronunciations based on the application of the pronunciation rules. Phonological variations across word boundaries are more difficult to handle. Consider the phrase "Did you see it?." The addition of the word /j u/ for the alternative pronunciation of "you", may conflict with the recognition of "judge".

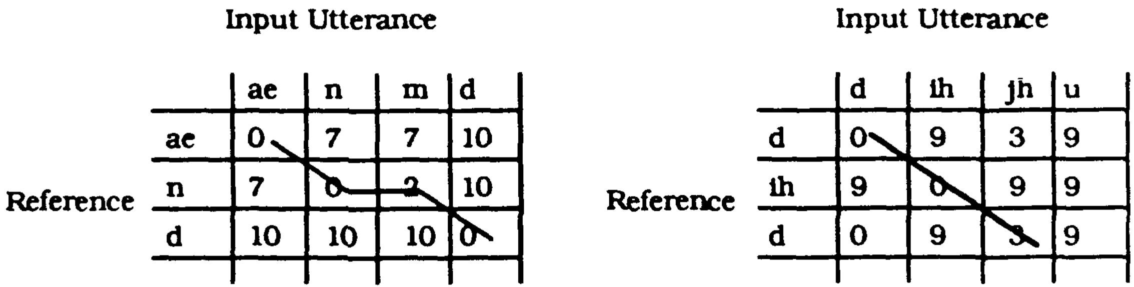
Another problem in interpreting phonetic strings is a single phonetic string with multiple interpretations. The following two sentences are a classic demonstration of this:

"Remember, a spoken sentence often contains many words that were not intended to be heard."
"Ream ember, us poke can cent tense off in contains men knee words that were knot in tend did tube bee herd."

Matching entries against a lexicon of representative transcriptions cannot resolve this problem. Other knowledge about syntax, semantics, and acoustic clues must be used to solve this problem. This problem was beyond the scope of our DTW word hypothesizer.

DTW has most commonly been applied to speech as the comparison of two time varying sequences of acoustic feature vectors. Each sequence defines the axis of a matrix mapping the feature vectors (per unit time) against one another. At each coordinate in the matrix a measure of distance or dissimilarity between the acoustic vectors is calculated. The goal is to find a path from the the first symbol in the feature vector to the last symbol whose cumulative distance (from the matrix values) is minimized. This method was extended to use phoneme strings as the unknown and reference patterns. The differences introduced by this extension relate to the time axis and the dissimilarity measures. Although the phoneme strings are time ordered each symbol may represent one or more arbitrary units of time. Therefore we are not strictly warping along a time axis. The difficulty with the dissimilarity measure was that there was no known metric for representing the difference between phonemes.

The matching algorithm traverses the matrix determining the least cost path. Figure 8-6 shows matching with an insertion error and with a deletion error.



Input Utterance

Reference	ae	n	m	d
ae	0	7	7	10
n	7	0	2	10
d	10	10	10	0

Input Utterance

Reference	d	ih	jh	u
d	0	9	3	9
ih	9	0	9	9
d	0	9	3	9

Figure 8-6 DTW matching algorithm

Searching all possible paths through the matrix is computationally expensive, so the search is constrained by limiting the degree of slope in the path, and setting a maximum permissible path distance. These constraints prune paths that would otherwise grow excessively large.

Ideally, a comprehensive inter-phoneme distance matrix (the phonetic dissimilarity measure) would be based upon the classification characteristics of the lower levels of the speech understanding system. At the time of the implementation only vowel-vowel confusability statistics where available. Distance data for consonants was extracted from studies of human confusability [SHEP80]. We still lacked any distance measure from vowels and consonants. We therefore assumed that in general the distance between consonants and vowels (excepting glides /y/ and /w/) was large enough to assume that their confusability was zero.

Discussion of the testing procedures and results for DTW word hypothesis are presented in section 8.5.6.

8.4.7 Natural Language Understanding using Conceptual Analysis

Our work in Natural Language Understanding has focused on the use of Conceptual Dependency (CD) theory as it is applied to cockpit-speech [RIDL89]. The goal of this work is to build a representation of meaning from the words in an utterance. CD [SCHA75, SCHA77, SCHA80] attempts to represent events as a composition of primitive actions, intermediate states, and causal relationships.

In this way, the utterances "Flight command gave F16#1 the target positions" and "F16#1 received the target positions from flight command" both could use the same representation. The representation would have the power to express that target positions were transferred from flight command to F16#1 and that target position information originated from flight command. The only significant difference between the two sentences is which actor initiated the transaction.

The language that fighter aircraft pilots use to communicate with each other about their actions, environment, and intentions, is very challenging language for Natural Language Understanding (NLU) systems to handle. This is because cockpit speech has evolved in a highly constrained environment and must meet different criteria of expressive power, legibility, syntax, and semantic content from every-day conversational language. The language that a person uses is *always* dependent on that person's environment. Cockpit language is more alien to the non-military, non-flying public than many of the ordinary variations in language because the fighter aircraft environment is so different from the normal day-to-day interactive environment of most people. The public is not, in general, acquainted with the cockpit of a fighter aircraft so the language used there appears to have little in common with the standard, everyday use of English.

There are many factors that have contributed to the unique language used by fighter pilots. A fighter aircraft is a technologically complex machine, so there tends to be a large amount of jargon and technical words surrounding its operation. Fighter aircraft are used primarily in the environment of the military, so there exists some military jargon not directly related to fighter aircraft. The military uses acronyms more heavily than the civilian population, so cockpit speech is also rife with acronyms. Another strong factor in shaping cockpit speech is the speed required for pilot communication. Events transpire very quickly in fighter aircraft, especially during battle situations, so the utterances must be short while maintaining a high information content. In sacrificing length for speed, one of the first casualties is *correct* English grammar. Only enough grammar remains to remove ambiguity of the roles the words play when combined with world knowledge and the present situation. Since connectives and qualifiers do not add as much information as they do length, they are left off except when absolutely necessary. The following utterances are from a cockpit speech, natural language study done by the Air Force. The utterances from this study form the domain for this thesis work.

Arm em up
Pass data
Pigeons to alternate
Gimme sidewinder
Select best weapon
Chaff and flares
How we doing
Radar enter track while scan target helicopter

The utterances above reflect the style of cockpit speech. They are compact and contain primarily verbs and nouns or, if a verb can be assumed, only nouns. These utterances demonstrate a dependence upon situation and environment to fill in missing information. For example, "Chaff and flares" may be a request to test these weapons, arm these weapons, or fire these weapons based on whether the pilot is performing systems checks, preparing for engagement, or already engaged with an enemy.

At the core of any representation of meaning, there must be a system for representing concepts. These concepts may be physical events, mental processes, causal ideas, statements of intent, as well as many other actions that may be described using natural language. The Conceptual Dependency (CD) theory use the following rule [SCHA75] for representing concepts.

Conceptualizations have the following
an ACTOR—the doer or performer of an ACT
an ACT—An action done to an object
an OBJECT—The thing that is acted upon
a RECIPIENT—The receiver of an object as the result of an ACT
a DIRECTION—in which that action is oriented
a STATE—the state

CD uses this same framework to represent concepts. If any of the required elements in the framework are missing, they must be postulated from knowledge sources other than the utterance being parsed. Such sources include: domain specific knowledge, knowledge of speaker habits, and previously parsed information.

The process which builds the conceptualization is the conceptual analyzer. The conceptual analyzer takes a sequence of words and builds the conceptualization indicated by the words. It uses a conceptual dictionary where the concepts that represent objects and actions within the domain are stored. In the domain of cockpit speech, the job of the analyzer is made more difficult by the extremely loose grammar, high use of contextual and domain knowledge, and the short utterances. The words in the utterance provide only enough information to disambiguate the meaning.

CD was chosen since it attempts to build meaning representation directly from the words that it sees. Also, since CD can be implemented as a frame-based, slot-filling mechanism the analysis of the utterance need not be done in a left-right fashion, but can instead be driven by expressive strength of the words that it sees. Thus the analyzer could start with the verb(s) in the utterance since they generally carry much of the information content of the utterance. From the conceptualization of the verb, objects can be postulated from the utterance or domain specific knowledge. It was felt that this was a good approach for handling the unique problems presented by the cockpit speech domain. One of the difficulties in applying CD to cockpit speech was that CD theory was developed to perform story understanding. These stories consisted of third person descriptions of human events. The man-machine interface application of CD which is demanded in cockpit-speech led to the addition of a few more primitive actions as well as methods for representing intelligent aircraft (machines with ability to understand and manipulate their environment in the manner of humans).

At the time of the writing of this report, this work was not yet at the implementation phase. In section 8.5.7 some conceptualizations for utterances from the domain will be presented.

8.5 Results of Speech Understanding Research

Attempting to report concisely the results of several years of work is a difficult task at best. We have decided to collect the results of each of the methodologies discussed above into this section so that they might be more easily referenced and reproduced. Each sub-section will briefly restate the the problem being addressed, present results, and finally some discussion and conclusions.

8.5.1 Feature Extraction (Formant Tracking) Results

The innovative work addressing low-level feature extraction was done in the area of formant tracking. We developed a statistical approach to formant tracking. This approach assigned probability measures to sets of features extracted from short-time analysis of the signal and a conditional mean estimate is used to determine format frequency values. This work was a generalization of methods introduced by Kopec based on hidden Markov models and vector quantization symbols.

Two approaches for calculating the probability measure were evaluated. The first approach calculated the probability of a formant occurring at a given time at given frequency from the classification based on the class distances calculated by the maximum likelihood classifier. This classifier is denoted FSD. The second method, computes probabilities from the observation frequencies of vector quantization symbols assigned to feature vectors. These observation frequencies are collected from training data. This classifier is denoted by FVQ. The FVQ work was included primarily for direct comparison to the work of Kopec.

	RMS Error			% Large Errors		
	F1	F2	F3	F1	F2	F3
Markel	92	274	503	1.2	5.4	21.3
S_{40}(1024)	72	93	150	1.6	0.4	1.5
S_{40}(64)	98	155	235	3.2	1.3	4.4
FSD	68	122	214	0.7	0.4	3.9
FVQ	99	154	224	3.7	1.1	4.8

Table 8-2 Formant Tracker Comparisons

Table 8-2 compares the FSD and FVQ to other types of formant trackers. Markel is a peak picking algorithm tested on the CMU database. S_{40}(1024) and S_{40}(64) are trackers used by Kopec with vector quantization codebook sizes of 1024 and 64 respectively using the TI connected digit database. In the table, smaller numbers indicate better performance. Large errors are defined by Kopec as a frame in which the absolute difference between the hand-marked value and the tracked value is 250Hz or greater in the case of F1 and 500Hz or greater in the cases of F2 and F3. The statistical trackers were all measured with a threshold value of 0.5

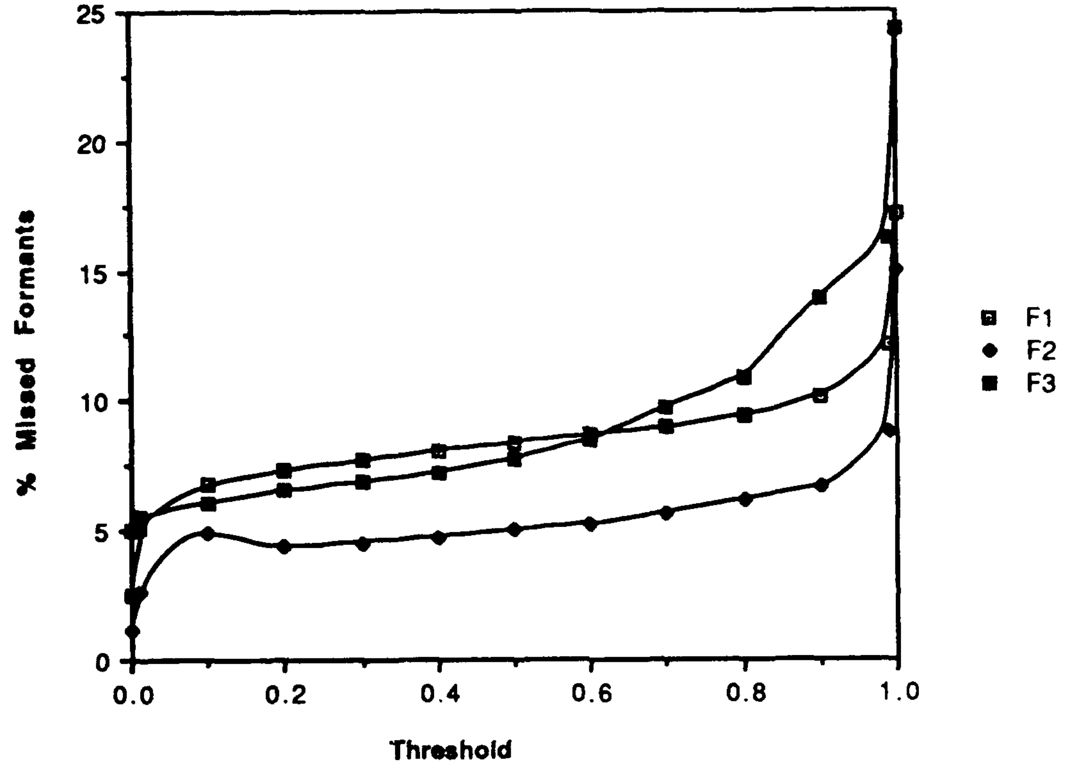


Figure 8-7 FSD Missed Formants

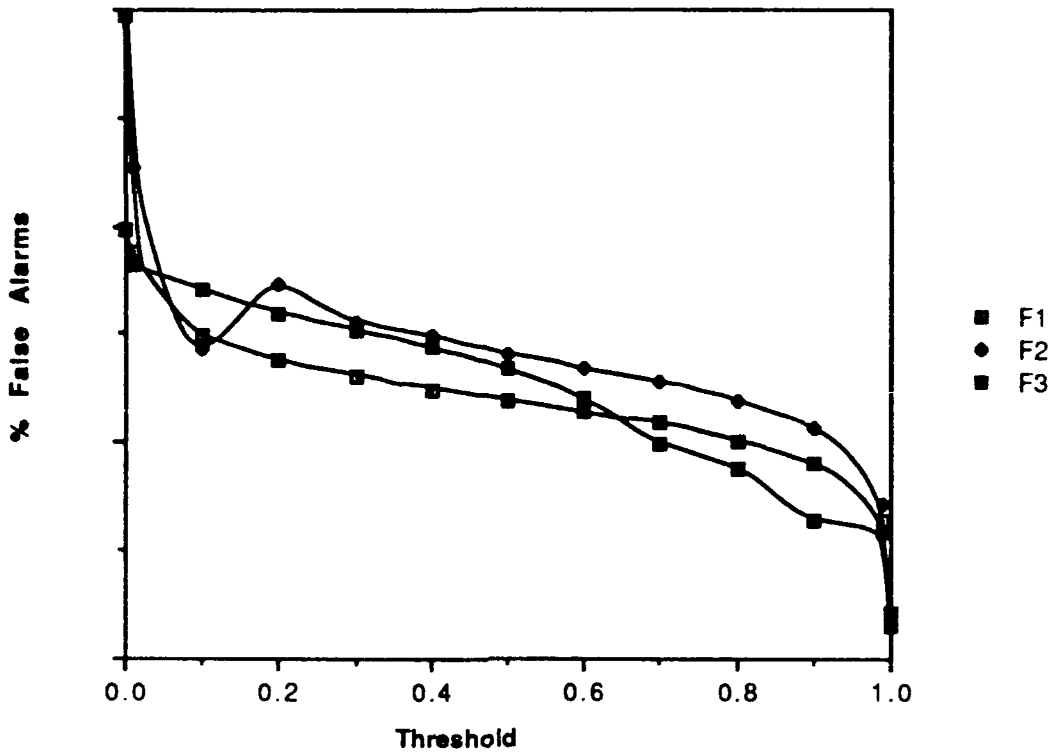


Figure 8-8 FSD False Alarms

Figures 8-7 and 8-8 show the FSD trackers performance with respect to missed formants and false alarms. A missed formant is defined as a frame in which there was a hand-marked value but no tracked value. Similarly, a false alarm is a frame in which there was a tracked value but no hand-marked value. The threshold value in the two figures represents the minimum probability necessary to predict a formant. Small thresholds will predict more formants but have a correspondingly high false alarm rate. Large thresholds have the opposite behavior.

The only approach to formant tracking in the literature which has been quantitatively analyzed in adequate fashion are Kopec's vector quantization trackers. In Kopec's study, approximately 142,000 frames were-hand marked. This is about four times as much data as we had available. The codebooks used by Kopec were trained on about 250,000 frames or about 12 times the training we used with the FVQ tracker. Referring back to table 8-2, we see that the peak picking algorithm scored well on F1, but poorly on F2 and F3. The high percentage of large errors, particularly for F3, reveals the difficulty in accurately assigning formant labels to spectral peaks. The results for the FVQ and $S_{40}(64)$ trackers were quite similar for all three formants. The FVQ codebooks may, however, have been insufficiently trained. The performance of the FSD tracker could be improved by allowing a more flexible configuration. The quantization size and number of features for each of the formants could be set independently, rather than using the same configuration for F1, F2 and F3. In particular, F3 would probably benefit from an increase in the number of features used since it has the largest frequency range and the worst performance.

8.5.2 Coarse Phonetic Classification Results.

The coarse phonetic categories studied were: vowel-like, strong fricative, weak fricative and silence. Two decision making procedures were investigated. The first procedure used Euclidean distance measure to clusters in the n-feature dimensional space that had been derived using the K-means clustering algorithm. The second procedure investigated the use of a multivariate maximum likelihood distance measure to classify the segments. Also investigated was the impact of the structure of the decision making process. Several tree-structured decision architectures were compared for each of the classification methods.

	Error Type			
Method	1	2	3	4
Max. Like. Single Level	68%	73%	86%	46%
Max. Like. Binary Tree	80%	84%	91%	58%
Max. Like. Skewed Binary	75%	81%	85%	54%
K-means Single Level	76%	84%	87%	48%
K-means Binary Tree	79%	83%	91%	49%
K-means Skewed Binary	78%	83%	90%	56%

Table 8-3 *Coarse Phonetic Classification Results*

In order to provide enough training data for the maximum likelihood classifier, all of the 96 utterances from the CMU data were used. Therefor, the results in table 8-3 represent training and testing on the same data set. In the table, the error types are as follows: 1.) Frames labeled correct out of all frames; 2.) Frames labeled correct disregarding frames at the boundaries of two phonetic classes; 3.) Segments correct to within 10ms of the boundary; 4.) Segments correct everywhere within the true segment. A segment is a sequence of frames all having the same class. For example, an 80ms noise associated with the phoneme /s/ would create a strong fricative segment of 8 frames. Error type 3 indicates an instance where any frame in the segment was correctly classified. Error type 4 indicates an instance where all frames except boundary frames in the segment where correctly classified. Error type 4 is the most stringent of the error constraints.

The best overall classifier was the maximum likelihood binary tree method. It was the top performer, or tied for the top, in all four error analysis methods. Of the tree structures, the binary tree performed best since the two systems using it outperformed all of the other system. These results indicate that the overall performance of both the classifiers was improved through the use of a decision tree structure and that the binary tree structure produced more accurate decisions than the other tree structures.

Possible improvements to the system would involve designing a second level of software to smooth the classes. The current system operates strictly on a frame-by-frame basis, taking almost no account of the characteristics of its neighboring frames. The only constraint involving neighboring frames is that a single frame surrounded on both sides by different frames will be changed to conform to the neighbors. HMMs are a very attractive method for performing the second level smoothing described above.

8.5.3 Fricative Expert System Results.

In our investigation of fricatives we examined the use of an expert system to classify fricatives based on expert spectrogram readers. The testing of the the expert system was limited by the SUN system upon which the system was running at the time. The system had been operating at about 97% capacity for some time. These space constraints allowed only 43 fricatives from four speakers (three male and one female) to be examined. The identification was considered correct when the system classified the sound a single phoneme which agreed with expert's opinion.

Correct identification was made 60% of the time. Included as incorrect were case where the system did not have enough knowledge to decide between competing candidates and therefore gave two choices as to the identity of the phoneme. Of the identifications considered incorrect, 41% were classifications that gave correct information about the segment but did not identify it as a single phoneme. Results for male speakers were substantially higher than for female speakers. 74% for males and 37% for females. Most of the errors that were made with female voices were due to a poor initial diagnosis of the amount of energy in the speech signal.

The following is an excerpt from an the expert-system dialog with the user: [user input in **bold** face]

```
Does the sound segment have
a) concentrations of energy mid or high range?
b) very diffuse energy
c) too little energy to tell
> a

Is there
a) a concentration of energy about 5kHz
b) a concentration of energy about 3kHz
c) both of these a) and b)
d) neither of these [a,b,c,d]
> why

******************************************
An investigation of the energy concentration is being performed
in order to discriminate between the s (alveolar) and sh (palatal)
sounds
```

```
c)ont e)lab h)elp :> c
*********************************************

Is there
a) a concentration of energy about 5kHz
b) a concentration of energy about 3kHz
c) both of these a) and b)
d) neither of these [a,b,c,d]
> a

place is alveolar
is there a voice bar present? [yes,no] > n
is there evidence of periodicity [yes,no] >n

Advice: could be s as in see

(Return continues) why

*********************************************
Since the answer to "Is there evidence of periodicity ?" is noev-
idence when the answer to "Is there a voice bar present ?" is
no. It follows that voicing is absent

Since the reply to "strong" is a) it follows that the place of ar-
ticulation is alveolar.

Since voicing is absent when the place of articulation is alveolar
advise "could be s as in see" in order to classify the fricative.
*********************************************
```

Enhancements to this work include re-working of the C program which initially determines the energy of the fricative. Of the incorrect identifications 41% were strong fricatives that were misclassified as weak by the C routine. Clearly the decision constraints in this process need adjustment. Most of the errors made with female voices were due to an initial misdiagnosis of the strength of the amount of energy in the signal by the user of the system. This may be due to the fact that the spectrograms being used measure spectral amplitudes to 6Hz, but many female voices have ranges up to 8kHz. Since strong fricatives have a concentration of energy in the high frequencies of the speaker's range, it is possible that the concentrations expected for strong fricatives was being lost.

8.5.4 Stop Consonant Classification Results

Our work in stop consonant classification is based on the shape of spectra taken from the burst region of the stop. Our general approach to stop classification has been the following: (1) Spectral Analysis of the stop is performed; (2) Feature Extraction from the spectra; (3) Optional feature compression; (4) Maximum likelihood classification.

The stop consonant classification testing is incomplete at the time of this writing. The results presented here represent the best classification results achieved by training and testing on all the stop consonants present in the vowel dense CMU database.

	/b/	/d/	/g/	/p/	/t/	/k/	Total
Closure							
2 frames 192pts	100	96.0	—	100	95.2	97.9	97.6
3 frames 192pts	—	100	—	100	100	100	100
No Closure							
2 frames 128pts	—	100	—	—	100		100
Combined							
4 frames 192pts	100	95.7	—	100	95.8	98.3	97.4

Table 8-4 Stop Consonant Classification (% correct)

The effect of not enough training data is evident in the classification results. Table entries with no values represent consonants that the maximum likelihood classifier could not establish mean feature vector values for, because there were not enough presentations of that particular stop consonant. Note that in stops not preceded by a closure only /d/ and /t/ could be classified. Work under development is being done with the combination of the CMU vowel dense and stop dense data. Final correct classification results are not available yet, but enough training data has been collected to eliminate this lost class problem. The results above represent the best combination of number of frames, frame width (in data points), and features when trained and tested against the vowel dense database. The following two tables show the classification results of place of articulation and voicing under the same testing conditions as above.

	Voiced	Unvoiced	Total
Closure			
4 frames 192pts	96.7	96.6	96.6
No Closure			
4 frames 128pts	97.4	100	99.1
Combined			
4 frames 256pts	88.9	93.7	92.0

Table 8-5 Stop Voicing Classification (% correct)

	Labial	Alveolar	Velar	Total
Closure				
3 frames 192pts	96.0	95.5	100	97.2
4 frames 256pts	96.7	96.6	100	96.6
No Closure				
1 frames 192pts	100	88.2	100	92.0
2 frames 128pts	100	100		100
Combined				
4 frames 192pts	95.7	93.0	96.1	94.5

Table 8-6 Stop Voicing Classification (% correct)

The work that is currently under investigation is aimed at optimizing the feature sets that are being used in order to reduce the *noise* (interfering features) that is presented to the maximum likelihood classifier. Work is also being done on the combined vowel dense and stop dense data which should allow better classification results (due to more training data) and training and testing against different data sets. Classification results for different training and testing sets are expected to be in the 70% range.

8.5.5 Vowel Classification Results

The performance of two types of classifiers was investigated for use with vowel and vowel-like segments. The first classifier was a maximum likelihood classifier similar to the classifier used in the coarse phonetic classification. The second classifier was a back-propagation neural network. These classifiers produced a phonetic classification for each 5ms frame of the analyzed vowel-like segment. This output sequence of phonemes was then presented to a hidden Markov model (HMM) which produced the final output string. The neural network and maximum likelihood classifiers might better be called pre-classifiers as they attempted to reduce the complexity of the classification performed by the HMM.

The features that were used to classify the vowels consisted of the first three formant frequency values, (F1, F2, F3), pitch (F0), mean (M2), variance (M2), skewness (M3), kurtosis (m4), median (m5), and the mean formant frequency values for each individual speaker (mF1, mF2, mF3). The more successful classification feature set included F1 and F2. These formant frequencies best characterize tongue height and advancement. This enables a distinction to be made between front, central, and back vowels.

Classification by the neural network and maximum likelihood pre-classifiers occurred at a frame by frame level of the vowel token at 5ms intervals. The database was split in half with each half having an equal distribution of tokens. One half was used for training the other for testing. The results achieved for a NN with all twelve features, 18 nodes in one hidden layer and momentum and gain terms of 0.5 and 0.3 produced training accuracy of 64.46% and testing accuracy of 47.38% as shown tables 8-7 and 8-8.

	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
iy	**85.66**	7.47	0.98	1.77	0.20	1.96	0.79	0.39	–	0.79
ih	9.37	**66.94**	5.51	4.41	3.03	1.65	6.34	0.55	1.10	1.10
eh	3.35	6.71	**47.87**	23.48	5.49	3.05	4.88	3.35	0.91	0.91
ae	2.21	5.22	13.25	**67.47**	2.41	0.80	1.41	5.62	1.00	0.60
ah	2.93	4.60	7.53	5.86	**61.09**	2.09	6.28	5.44	2.51	1.67
uw	8.21	8.70	1.45	2.90	1.45	**66.18**	8.70	0.48	0.48	0.97
uh	5.10	13.27	14.29	6.12	9.69	12.76	**32.14**	2.55	3.57	0.51
aa	2.17	3.26	5.98	21.74	8.15	0.54	4.89	**47.83**	3.26	2.17
ao	0.90	2.70	13.51	14.41	6.31	–	9.91	9.91	**40.54**	1.80
er	2.33	2.33	2.33	0.93	1.86	0.47	1.86	0.47	0.93	**86.51**
Average rate of correct decisions						64.46				

Table 8-7 Training Results for NN Vowel Classifier

	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
iy	**75.92**	13.29	0.77	–	0.39	6.94	1.93	0.77	–	–
ih	14.50	**62.21**	3.44	1.53	2.29	12.60	3.44	–	–	–
eh	2.17	11.15	**26.32**	26.01	12.38	1.86	17.96	–	–	2.17
ae	–	8.60	13.63	**63.73**	6.50	–	2.94	3.98	–	0.63
ah	–	14.80	5.16	15.49	**14.08**	–	46.48	3.76	–	0.94
uw	7.63	3.21	–	–	0.40	**48.59**	38.96	–	–	1.20
uh	–	5.77	–	–	12.98	29.33	**47.60**	3.37	–	0.96
aa	–	4.62	4.62	21.97	42.77	0.58	18.50	**4.62**	–	2.31
ao	–	–	–	21.85	31.93	–	32.77	–	–	13.45
er	–	9.72	–	11.11	12.50	1.39	14.58	2.78	–	**47.92**
Average rate of correct decisions						47.38				

Table 8-8 Testing Results for NN Vowel Classifier

Training the maximum likelihood pre-classifier was managed in a slightly different fashion. An exhaustive search of which combination of N features that best the vowel tokens was performed, where N ranged between 3 and 8. Pre-classification results of the optimal feature set are shown in table 8-9.

Features Used	% Correct
F_1,F_2,F_3	55.68
F_1,F_2,F_3,F0	61.82
F_1,F_2,F_3,F0,mF_2	64.89
F_1,F_2,F_3,F0,mF_2,M5	66.99
F_1,F_2,F_3,F0,mF_2,M5,mF_3	68.54

Table 8-9 Gaussian preclassification results based upon feature set

The feature set finally used for the maximum likelihood classifier was F1, F2, F3, mF1, mF3 based upon its overall performance. Table 8-10 shows the training results of the classifier. The jack-knife statistic shown in table 8-10 is obtained by removing the individual influences of the training tokens from the training statistics, thus giving a more accurate indication of what the testing performance might be.

	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
iy	**82.88**	8.37	0.88	0.29	0.78	4.77	0.19	–	–	1.85
ih	11.52	**60.80**	7.04	–	4.96	10.24	4.0	0.16	–	1.28
eh	1.38	9.22	**51.0**	17.36	8.29	0.92	2.76	3.53	2.92	2.61
ae	1.33	3.69	16.82	**60.21**	3.59	0.10	0.41	12.21	0.41	1.23
ah	0.44	4.87	5.31	3.10	**29.65**	3.32	16.81	23.45	11.95	1.11
uw	7.02	1.54	–	0.22	1.32	**71.93**	15.13	–	–	2.85
uh	–	4.21	5.45	–	10.15	16.58	**55.45**	1.73	4.70	1.73
aa	0.84	2.52	2.52	9.52	15.13	–	1.12	**56.02**	10.64	1.68
ao	–	–	0.87	2.17	0.43	2.61	5.65	–	**88.26**	–
er	0.56	0.84	1.39	–	–	3.06	1.11	0.84	1.67	**90.53**
Total	.18	.11	.11	.14	.07	.10	.08	.08	.06	.07
Average rate of correct decisions						64.39				
Jack-knife						54.92				

Table 8-10 Training results for a Gaussian pre-classifier

The final part of the study involved taking the results of the pre-classifiers and classifying the vowels over time with the aid of a HMM. A three state left to right model was constructed for every vowel class that was being investigated. The results of applying the HMM to the phonetic pathway indicated by the backpropagation neural network appear in table 8-11

	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
iy	**89.06**	4.68	–	–	–	6.25	–	–	–	–
ih	4.65	**76.74**	4.65	–	2.33	4.65	4.65	2.33	–	–
eh	–	5.0	**65.00**	12.50	5.0	2.50	2.50	5.0	–	2.50
ae	–	2.50	7.50	**80.00**	2.50	–	–	7.50	–	–
ah	–	9.68	9.68	3.23	**48.39**	–	16.13	12.90	–	–
uw	4.0	8.0	4.0	–	–	**56.00**	24.00	–	–	4.0
uh	–	12.0	12.0	–	7.14	17.86	**50.00**	3.57	–	–
aa	–	5.88	5.88	5.88	29.41	–	5.88	**52.94**	–	–
ao	–	–	–	–	30.00	–	–	40.00	**20.00**	10.00
er	–	6.66	–	6.66	–	–	–	13.33	–	**73.33**
Jack knife						68.05				

Table 8-11 Results of hidden Markov model with neural network

The results of applying the HMM vowel models to the output of the maximum likelihood pre-classifier are presented in table 8-12.

	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
iy	**89.06**	4.69	–	3.12	–	1.56	–	–	–	1.56
ih	9.30	**60.47**	9.30	–	2.33	11.63	4.65	2.33	–	–
eh	–	7.50	**50.00**	20.00	5.00	2.50	5.00	7.50	–	2.50
ae	–	2.50	17.50	**65.00**	2.50	–	–	12.50	–	–
ah	–	6.45	3.23	–	**25.81**	3.23	19.35	32.26	6.45	3.23
uw	4.00	4.00	4.00	–	–	**64.00**	16.00	4.00	–	4.00
uh	–	–	7.14	–	10.71	21.43	**50.00**	–	7.14	3.57
aa	–	–	5.88	5.88	11.76	–	–	**58.82**	11.76	5.88
ao	–	–	–	–	10.00	10.00	–	10.00	**70.00**	–
er	–	–	–	–	–	–	–	–	–	100.0
Total	.20	.12	.12	.12	.06	.10	.09	.10	.04	.07
Average rate of correct decisions						63.58				
Jack-knife						58.15				

Table 8-12 Results of hidden Markov model with Gaussian classifier

As shown in previous studies [HILL87], formant frequency values tend to be the best features that characterize a vowel. The drawback in using these values resides in the difficulty of accurately tracking frequency values. The comparison of the maximum likelihood method and neural network is inconclusive for this study. Each model has its own strengths and weaknesses. A neural network tends to be more forgiving when presented irrelevant features, whereas the Gaussian model is apt to "memorize" the insignificant features and thus lose its ability to generalize about the specific class the training token came from. On the other hand the time required to train a neural network takes a few hours in contrast to the few minutes needed to train the maximum likelihood classifier.

Further work in this area includes the investigation different types of neural network models, and the development of a more speaker independent feature. The models should not have to rely on the normalizing values of mF1, mF2, and mF3.

8.5.6 Word Hypothesis from Errorful Phonetic Strings Results

This project investigated a dynamic programming approach to the word hypothesis problem. It was based on an approach known as Dynamic Time Warping (DTW). DTW is a common method of sequence comparison used in matching a reference vector with an unknown vector. As applied to phonetic strings, the DTW algorithm compared the unknown phonetic string with reference strings from a database of words. A cumulative least cost path combined with an empirically determined threshold was used as the decision criteria for recognition.

A comprehensive series of tests were run against a group of forty-two phrases containing ten-percent substitution errors. Of insertion, deletion, and substitution errors, substitution errors are the most common in speech and caused less problems for the DTW than the other two. Each test varied one of three primary variables: the minimum distance threshold, the number of candidate words accepted at any one phonetic index, and the size of the lexical search space when obtaining reference patterns for the DTW process. Five threshold values were used ranging from zero to 1000. Values ranging from five to thirty were used as the number of candidates accepted at any index. The *small* search space provided access to approximately twenty to thirty percent of the word-initial phoneme groups in the lexicon, whereas the large search space was twice that.

The performance of each test was evaluated using the following criteria. First, the final word lattice returned from each parsing was examined for the presence of the intended utterance. Finding all utterance words in their correct order was considered a complete match. The number of complete matches from N test utterances provided the total percent recognition. A second measure of success was the average percentage of words hypothesized per phrase. This gives a relative idea of how well the parsing process is working on a phrase basis. It relates the number of correct words found to the number in the original utterance over all phrases.

The computational speeds for the process were disappointing. On an Explorer I, times ranged from 4 to 21 hours per test (42 utterances). Equivalent tests on the Explorer II reduced these times by a factor of five. There was a positive correlation between increased run times and increases in all three variables. Table 8-13, shows the range of recognition rates for utterances in the test suite.

Threshold	Low Recognition Rate	High Recognition Rate
0	36	50
125	46	50
175	48	50
225	57	63
275	57	66
500	63	66
1000	63	66

Table 8-13 Recognition Rates for DTW algorithm

Results from the most successful test were selected for more through examination. An immediate observation was that in most cases, words not hypothesized were of short transcription length (under four phonemes). Aware that there are several points within the hypothesis procedure that a potential candidate may be pruned before acceptance, the missed words were submitted individually for parsing and monitored as to when they were dropped. It was discovered that the DTW comparison procedure was penalizing small words severely and pruning them early on. Words with significantly different transcription lengths should be treated differently. As reference pattern length grows, the distance generated by a single mis-match in the warping process has less impact on possible rejection due to averaging. Therefor, the decision to prune shorter words based on distance must be made earlier than longer words. After making this adjustment to the process, total phrase recognition rates increased by fourteen percent to a high of 80%.

From various veiwpoints, the dynamic time warping method of this study has proven to be not entirely satisfactory for use in the hypothesis of words from errorful phonetic strings. Three minute per test phrase run-times are magnitudes away from real time processing. It was demonstrated that noise levels (measured as the total number of words required to identify a single correct word) in the test with greatest recognition rate approached ninety to one. This places a tremendous burden on higher level processes which must detect the correct phrase from the word lattice. Assuming that twenty were hypothesized at each of ten phonetic indices, there are 20^{10} possible phrases in which to find the actual phrase. For these reasons alternative methods must be investigated for word hypothesis in continous speech. given a different organization or additional information within the lexicon, syntactic and semantic constraints might be used to selectively predict which words (reference patterns) are most likely to be present in the unknown utterance. Hidden Markov models are also an attractive alternative.

8.5.7 Natural Language Understanding using Conceptual Analysis Results

Our work in Natural Language Understanding has focused on the use of Conceptual Dependency (CD) theory as it is applied to cockpit-speech. The goal of this work is to build a representation of meaning from the words in an utterance. Much of this project is still in the design area. Two diagrams will be presented showing the type of behavior expected from the completed project.

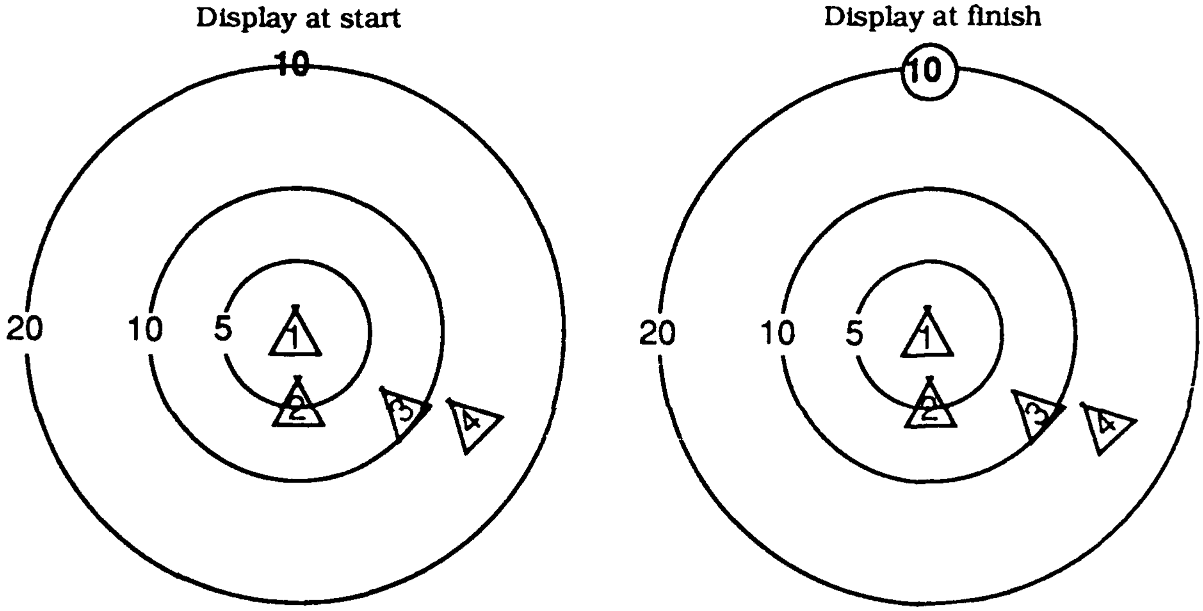
Figure 8-9 is from the cockpit-speech natural language study and is provided as an indication of the use of natural language as a man-machine interface to the aircraft. It also demonstrates the unique qualities of cockpit speech.

Description:
Pilot is approaching rendezvous. A data link message has been receive concerning down range threats.

Voice from aircraft before command:
"Threat data."

Experimenter's prompt for command:
"request details on threat data."

Voice response from aircraft after command
"Tracking J band."


Display at start

Display at finish

Utterances Issued:
threat status, radar sort, is it in an active mode, threat data is it a threat, go, evasive course,
am i targeted, display information, threat stats, defeat, more data, analyze, geve me the threat data,
present the data, give me more inforamtion on the threat, threat ring, go data, describe threat,
give me a threat ring, go data, is the site active, give me the data, threat locked on to me, go ahead,
status ten, sam zone show me the threat.

Figure 8-9 Cockpit Speech Domain
Adapted from [LIZZ87]

Figure 8-10 is graphical representation of the conceptualization for the situation described above. Of note are a new primitive action, DISPLAY, which takes image information as an object and presents it on some type of visual device. It is analogous to the human action of speaking. The conceptualization of threat data is a list of attributes which make up the object. These attributes have not yet been finalized.

1.3.02 "Request details on threat data"

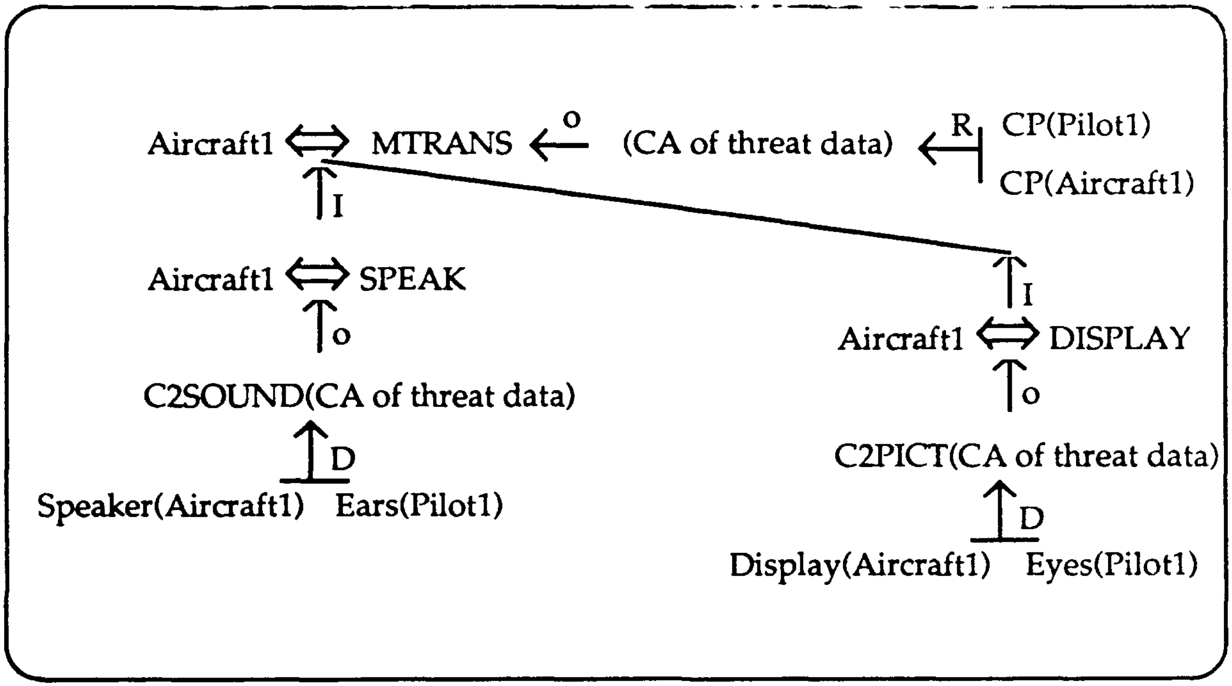

Aircraft1 ⇔ MTRANS ← O (CA of threat data) ← R CP(Pilot1)
CP(Aircraft1)
I
Aircraft1 ⇔ SPEAK
O
C2SOUND(CA of threat data)
D
Speaker(Aircraft1) Ears(Pilot1)
I
Aircraft1 ⇔ DISPLAY
O
C2PICT(CA of threat data)
D
Display(Aircraft1) Eyes(Pilot1)

Figure 8-10 Example of CD conceptualization

The majority of the phrases in figure 8-9 should map onto the meaning representation of figure 8-10.

The work in this area is challenging, but there is hope for some interesting results, particularly in determining how to address the unique problems presented by cockpit speech.

8.5.8 NAIC Ancillary results

Aside from the speech understanding project, we have supported at RIT, the educational and technical goals of expanding the base of AI research. This is best demonstrated by comparing the status of AI at RIT in 1984 with that in 1989.

The faculty with AI expertise has increased from 1 to 9. This includes computer science faculty, faculty from other disciplines at RIT, and the full-time employees at RIT Research Corp's Intelligent Systems Division. The AI curriculum in 1984 consisted of a single survey type course. As of 1989 there are both graduate and undergraduate concentrations in AI as well as many one-time seminars. By 1984 5 AI theses had been completed. That number has grown to 60.

In 1984 there was no funded research for AI at RIT. Today the following projects have been funded (NAIC—Speech Understanding, RADC & TI—Speech Workstation Development, RADC—Intelligent Surveillance System (with Buffalo & RPI), US Govt.—Intelligent Signal Processing, US Govt.—Object Oriented Simulation, US Govt.—Intelligent Environment for Mission Planning, US Govt.—Neural Network Applications to Chinese Character Recognition, US Govt.—Smalltalk and Object-Oriented databases, Eastman Kodak—Neural Network Application to Signal Detection). We have built up a host of AI equipment including two Explorer II, 2 Explorer I, Sun 2/130, several Macintoshes running common LISP.

We have established an AI presence at RIT and are well prepared to study new applications and problems.

8.6 Conclusions and Implications for Further Development

The five-year NAIC Speech Understanding project has provide RIT the ability to study speech and the AI techniques that can be applied to speech. We feel that we now have a excellent understanding of the difficult problems encountered in developing a large vocabulary, continuous speech, speaker-independent, speech understanding system. We believe that we understand what the solutions to many of those problems are as well.

Future developments could include the development of a commercial quality speech understanding system based upon our prototype system, the incorporation of adaptive processes and learning into the system, and the testing and evaluation of as yet undiscovered speech understanding procedures.

Our work in the speech understanding area has allowed us to develop tools, technologies, and personnel that may be applied to other speech related disciplines. The speech understanding work we have done has extensions in the areas of: speaker identification, language identification, and key word spotting.

In summary, we have established a real AI research resource at RIT and need only new challenges and problems upon which to apply it.

8.7 References

[ATKI87] Atkinson, K.,, "FRIC — An Expert System to Recognize Fricatives", unpublished Masters Thesis, Rochester Institute of Technology, 1987.

[BLUM79] Blumstein S.E., and Stevens, K.N., "Acoustic Invariance in Speech Production: Evidence from Measurements of the Spectral Characteristics cf Stop Consonants", J. Acoust Soc of America, Vol66 #4, 1979.

[CAMP89] Campanelli, M., "Computer classification of phonetically segmented stop consonants in a speaker independent continuous speech environment", unpublished Masters Thesis under development, Rochester Institute of Technology, 1988.

[DELM88] Delmege, J., "CLASS — A study of methods for coarse phonetic classification", unpublished Masters Thesis, Rochester Institute of Technology, 1988.

[GAYV89] Gayvert, R., "A Statistical Approach to Formant Tracking", unpublished Masters Thesis, Rochester Institute of Technology, 1989.

[HILL87] Hillenbrand, J. and Gayvert, R.T., "Speaker-Independent Vowel Classification Based on Fundamental Frequency and Format Frequencies", Journal of the Acoustical Society of America, Spring 1987, 81 (Suppl. 1), S93 (A).

[ITAK75] Itakura, F., "Minimum Prediction Residual Principle Applied to Speech Recognition", IEEE Transactions Acoustical, Speech, Signal Processing, ASSP-23, 1975, 67-72.

[KEWL83] Kewley-Port, D., "A Time-varying features as correlates of place of articulation in stop consonants", J. Acoust Soc of America, Vol73 #1, 1983, 232-235.

[KOPE86] Kopec, G.,"Formant Tracking Using Hidden Markov Models and Vector Quantization", IEEE Trans., ASSP-34, 1986, 709-729

[LIZZ87] Lizza, Capt. G., Munger, M., Small, Capt. R., Feitshans, G., and Detro, S., "A Cockpit Natural Language Study - Data collection and Initial Data Analysis", Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, April 1987, Doc.#AFWL-TR-87-3003.

[MARK72] Markel J.D., "The SIFT Algorithm for Fundamental Frequency Estimation", IEEE Trans. Audio Electroacoustics, AU-20 5, 1972, 367-372.

[MARK76] Markel J.D. and Gray A.H. Jr., *Linear Prediction of Speech*, Springer-Verlag, 1976.

[PETE52] Peterson, G., and Barney, H., "Control Methods used in a Study of the Vowels", J. Acoust Soc of America, Vol 24, 1952, 175-184.

[RABI86] Rabiner, L.R. and Juang, B.H., "An Introduction to Hidden Markov Models", IEEE ASSP Magazine, January 1986, 4-16.

[RIDL89] Ridley, T., "Conceptual Analysis techniques applied to understand cockpit-speech", unpublished Masters Thesis under development, Rochester Institute of Technology, 1989.

[RITR84] RIT Research Corp. of Rochester Institute of Technology, "Proposal — RADC Artificial Intelligence Research Program", PRDA 84-01, July 1989.

[RITR89] RIT Research Corp. of Rochester Institute of Technology, "Final Report of the ESPRIT System", CDRL No. F003 of RADC/IRAA contract No. F30602-87-D-0090 Task I06, 1989.

[SCHA75] Schank, Roger C. *Conceptual Information Processing.* Amsterdam: North-Holland, 1975.

[SCHA77] Schank, Roger C., and Robert P. Abelson. *Scripts, Goals, Plans and Understanding.* Hillsdale, NJ: Lawrence Erlbaum, 1977.

[SCHA80] Schank, Roger C., and Christopher K. Riesbeck. *Inside Computer Understanding.* Hillsdale, NJ: Lawrence Erlbaum, 1980.

[SELL89] Sellman, R.T., "Word Hypothesis in Undifferentiated, Errorful Phonetic Strings", unpublished Masters Thesis, Rochester Institute of Technology, 1989.

[SHEP80] Shepard, R., "Multidimensional Scaling, Tree-Fitting, and Clustering", Science, 210, October 1980, 390-398.

[STEV78] Stevens, K.N. and Blumestein, S.E., "Invariant Cues for Place of Articulation in Stop Consonants", J Acoust Soc of America, Vol 64 #5, 1978, 1358-1368

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control, Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of C^3I systems. The areas of technical competence include communications, command and control, battle management information processing, surveillance sensors, intelligence data collection and handling, solid state sciences, electromagnetics, and propagation, and electronic reliability/maintainability and compatibility.